DEEP LEARNING-BASED APPROACHES FOR PLANT DISEASE AND WEED DETECTION

A THESIS BY PUBLICATIONS PRESENTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN

ENGINEERING

MASSEY UNIVERSITY, AUCKLAND, NEW ZEALAND

Muhammad Hammad Saleem

2022

Abstract

To match the ever-growing food demand, the scientific community has been actively focusing on addressing the various challenges faced by the agricultural sector. The major challenges are soil infertility, abrupt changes in climatic conditions, scarcity of water, untrained labor, emission of greenhouse gases, and many others. Moreover, plant diseases and weeds are two of the most important agricultural problems that reduce crop yield. Therefore, accurate detection of plant diseases and weeds is one of the essential operations to apply targeted and timely control measures. As a result, this can improve crop productivity, reduce the environmental effects and financial losses resulting from the excessive application of fungicide/herbicide spray on diseased plants/weeds. Among various ways of plant disease and weed detection, image-based methods are significantly effective for the interpretation of the distinct features. In recent years, image-based deep learning (DL) techniques have been reported in literature for the recognition of weeds and plant diseases. However, the full potential of DL has not yet been explored as most of the methods rely on modifications of the DL models for well-known and readily available datasets. The current studies lack in several ways, such as addressing various complex agricultural conditions, exploring several aspects of DL, and providing a systematic DL-based approach.

To address these research gaps, this thesis presents various DL-based methodologies and aims to improve the mean average precision (mAP) for the identification of diseases and weeds in several plant species. The research on plant disease recognition starts with a publicly available dataset called PlantVillage and comparative analyses are conducted on various DL feature extractors, meta-architectures, and optimization algorithms. Later, new datasets are generated from various local New Zealand horticultural farms, named NZDLPlantDisease-v1 & v2. The proposed datasets consist of healthy and diseased plant organs of 13 economically important horticultural crops of New Zealand, divided into 48 classes. A performance-optimized DL model and a transfer learning-based approach are proposed for the detection of plant diseases using curated datasets. The weed identification has been performed on an open-source dataset called DeepWeeds. A two-step weed detection pipeline is presented to show the performance improvement of the deep learning model with a significant margin. The results for both agricultural tasks achieve superior performance compared to the existing method/default settings. The research outcomes elaborate the practical aspects and extended potential of DL for selected agricultural applications. Therefore, this thesis is a benchmark step for cost-effective crop protection and site-specific weed management systems (SSWM).

Authors Declaration

This thesis was produced with eight journal publications according to Massey University's "Ph.D. thesis by publications" guidelines. This thesis is based on research that has been published. Four articles are published in MDPI, one in Springer, one in IEEE, and two in Frontiers journals. In accordance with MDPI, Springer, IEEE, and Frontiers' copyright policy, this thesis contains the accepted version of each published manuscript, rather than the corresponding final versions. Consequently, while the content is identical to the published versions, there may be stylistic differences between the enclosed work and the published versions. Furthermore, the work contained in this thesis was published in several different journals. Therefore, some of the submitted chapters are relatively succinct, there are some repetitions (particularly in the literature review of some chapters), and there are small stylistic differences between the chapters themselves.

Acknowledgments

Praise and thanks to Almighty Allah for blessing me with the strength to complete this thesis. I would like to thank my supervisor Khalid Mahmood Arif for his continuous guidance, support, and mentorship. His motivation, attention to my work, encouragement, and belief in me brought the best out in me. He guided me through everything to become a valuable part of the scientific community. His sincere advice has helped me to build my career in a better way. I am grateful to my co-supervisor Johan Potgieter for his valuable support and guidance throughout this Ph.D. journey. I am thankful to him for providing me with an opportunity to explore the horticultural fields in Palmerston North.

I cannot describe in words how grateful I am for my parents Muhammad Saleem and Sajida Saleem, without whom I could not have achieved anything in my life. Their prayers, guidance, support, and countless sacrifices for me and my career throughout their lives, have been the main source of any sort of success in my life. They are my inspiration and my real-life heroes. Their love, affection, and encouragement gave me the strength to endure tough times. I am always thankful to Allah to bless me with the best parents. My incredibly special thanks to my lovely wife Fakhia Hammad, whose love and support have no boundaries. She is not only my life partner but also a source of inspiration and motivation. Her numerous sacrifices have resulted in the completion of this work. She also boosted my confidence during my tough times, and being an engineer, technically supported me by asking various research-related questions and helping me illustrate figures for the research articles. I cannot miss my brothers, Muhammad Hassam and Muhammad Taha for always supporting me, praying for me, and always being my best friends.

I would like to thank Riaz Uddin (Associate Professor, NED University, Pakistan) for guiding me on various aspects of the Ph.D. journey. I am very grateful to all my friends and seniors, Muhammad Asif Ali Rehmani, Ayesha Shaukat, Swapna Jaywant, Rahila Umer, Muhammad Harris, Saad Aslam, and Sachin Katare. They helped me in various phases of my Ph.D., guided me, and shared their great experiences with me, which helped me throughout this journey. I am also grateful to Dianne Cook (Massey University, Auckland), Ruth Brooks (Massey University, Auckland), Louise Hopkins (MAF Digital Lab, Massey University, Palmerston North), and Russell Wilson (MAF Digital Lab, Massey University, Palmerston North) for providing various useful stuff and information throughout my study. Finally, I would like to acknowledge the Ministry of Business, Innovation, and Employment (MBIE), New Zealand, Science for Technological Innovation (SfTI) National Science Challenge, for funding my Ph.D. study at Massey University.

Publications

1. **M. H. Saleem**, J. Potgieter, and K. M. Arif, "Plant disease detection and classification by deep learning," *Plants*, vol. 8, no. 11, p. 468, 2019. https://doi.org/10.3390/plants8110468 Q1, IF: 4.658
   Award: *2021 Best paper award by the Selection Committee composed of Editorial Board Members of Plants MDPI journal.*
2. **M. H. Saleem**, J. Potgieter, and K. M. Arif, "Plant disease classification: A comparative evaluation of convolutional neural networks and deep learning optimizers," *Plants*, vol. 9, no. 10, p. 1319, 2020. https://doi.org/10.3390/plants9101319 Q1, IF: 4.658
3. **M. H. Saleem**, S. Khanchi, J. Potgieter, and K. M. Arif, "Image-based plant disease identification by deep learning meta-architectures," *Plants*, vol. 9, no. 11, p. 1451, 2020. https://doi.org/10.3390/plants9111451 Q1, IF: 4.658
4. **M. H. Saleem**, J. Potgieter, and K. M. Arif, "Automation in agriculture by machine and deep learning techniques: A review of recent developments," *Precision Agriculture*, vol. 22, no. 6, pp. 2053-2091, 2021. https://doi.org/10.1007/s11119-021-09806-x Q1, IF: 5.767
5. **M. H. Saleem**, K. K. Velayudhan, J. Potgieter, and K. M. Arif, "Weed identification by single-stage and two-stage neural networks: A study on the impact of image resizers and weights optimization algorithms," *Frontiers in Plant Science*, vol. 13, 2022. https://doi.org/10.3389/fpls.2022.850666 Q1, IF: 6.627
6. **M. H. Saleem**, J. Potgieter, and K. M. Arif, " Weed detection by Faster RCNN model: An enhanced anchor box approach," vol. 12, no. 7, p. 1580, *Agronomy*, 2022. https://doi.org/10.3390/agronomy12071580 Q1, IF: 3.949
7. **M. H. Saleem**, J. Potgieter, and K. M. Arif, "A performance-optimized deep learning-based plant disease detection approach for horticultural crops of New Zealand," *IEEE Access*, vol. 10, pp. 89798-89822, 2022. http://doi.org/10.1109/ACCESS.2022.3201104 Q1, IF: 3.367
8. **M. H. Saleem**, J. Potgieter, and K. M. Arif, "A weight optimization-based transfer learning approach for plant disease detection of New Zealand vegetables," *Frontiers in Plant Science*, vol. 13, 2022. https://doi.org/10.3389/fpls.2022.1008079 Q1, IF: 6.627

Table of Contents

Table of Contents

List of Tables

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Chapter 7

Chapter 8

List of Tables

Appendix 1

List of Figures

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Chapter 7

Chapter 8

Appendix 1

List of Figures

List of Abbreviations

Abbreviations used in the thesis are given below in alphabetic order:

2D-CNN-BidGRU	Two-Dimensional Convolutional Bidirectional Gated Recurrent Unit Neural Network
ADAM	Adaptive Moment Estimation
AE	Autoencoder
AI	Artificial Intelligence
ANN	Artificial Neural Network
AP	Average Precision
API	Application Programming Interface
BaRTs	Bagged Regression Trees
BoRTS	Boosted Regression Trees
BPNN	Back-Propagation Neural Network
CA	Classification Accuracy
CART	Classification and Regression Trees
CCP	Correct Classification Percentage
CNC	Computer Numerical Control
CNN	Convolution Neural Network
COCO	Common Objects In Context
DBN	Deep Belief Networks
DCNN	Deep Convolutional Neural Network
DCRN	Densely Connected Recurrent Convolutional Network
DL	Deep Learning
DoF	Degree-of-Freedom
DT	Decision Tree
EBT	Ensemble Boosted Tree
ELM	Extreme Learning Machine
ELU	Exponential Linear Unit
FC	Fully Connected
FCR	False Colour Removal
FRKNN	Feature Ranking K-Nearest Neighbor
GAN	Generative Adversarial Network
GA-SVM	Genetic Algorithm Support Vector Machine
GHG	Greenhouse Gases
GLCM	Grey-Level Co-Occurrence Matrix
GPDCNN	Global Pooling Dilated Convolutional Neural Network
GPU	Graphical Processing Unit
GRU	Gated Recurrent Unit
HSI	Hyperspectral Imaging
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
INAR-SSD	Single Shot Detector With Inception Module and Rainbow Concatenation
IoRT	Internet of Robotic Things
IoU	Intersection of Union
IRCNN	Inception Recurrent Convolutional Neural Network
IRRCNN	Inception Recurrent Residual Convolutional Neural Network
KNN	K-Nearest Neighbor

LDA	Linear Discriminant Analysis
LR	Logistic Regression
LRP	Layer-Wise Relevance Propagation
LSSVM	Least Squares Support Vector Machine
LSTM	Long Short-Term Memory
mAP	Mean Average Precision
MAV	Micro Aerial Vehicle
ML	Machine Learning
MLCNN	Multilayer Convolutional Neural Network
MLP	Multilayer Perceptron
MRF	Markov Random Field
MSER	Maximally Stable Extremal Regions
NiN	Network In Network
NMS	Non-Maximum Suppression
OC-SVM	One Class Support Vector Machine
OR-AC-GAN	Outlier Removal-Auxiliary Classifier-Generative Adversarial Nets
PCA	Principal Component Analysis
PSO	Particle Swarm Optimization
PSPNet	Pyramid Scene Parsing Network
PV	PlantVillage
R2U-Net	Recurrent Residual Convolutional Neural Network Based On U-Net Model
RBFNN	Radial Basis Function Neural Network
R-CNN/RCNN	Region-based Convolutional Neural Network
ReLU	Rectified Linear Unit
RF	Random Forest
R-FCN/RFCN	Region-based Fully Convolutional Networks
RMSProp	Root Mean Square Propagation
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
ROI	Region of Interest
RPN	Region Proposal Network
SGD	Stochastic Gradient Descent
SIRF	Scale-Invariant Feature Transform
SLIC	Simple Linear Iterative Clustering
SRCNN	Super-Resolution Convolutional Neural Network
SSD	Single-Shot Multibox Detector
SURF	Speeded Up Robust Features
SVM	Support Vector Machine
SVR	Support Vector Regression
TL	Transfer Learning
ToF	Time-of-Flight
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
VFH	Viewpoint Feature Histogram
YOLO	You Look Only Once

Chapter 1. Introduction

1.1. Background

The United Nations Organization reports that the world population is expected to reach 9.1 billion by 2050, and the food demand will increase by up to 70 percent [1]. In this regard, the agricultural sector plays a vital role in meeting the global food requirements. Furthermore, according to recent reports by the New Zealand Institute of Plant & Food Research Limited, the export value of horticultural crops has been growing over the past years. The export value has been recorded NZ$ 6.68 billion in 2021 [2] indicating the importance of the horticultural industry to the New Zealand economy. In the light of the current statistics (global and local), the scientific community has been actively seeking solutions to various real/complex problems related to agriculture by deploying efficient and intelligent systems [3].

In general, the output of an agricultural field depends on various factors, such as trained labor, climatic conditions, continuous monitoring of crops, soil moisture content, water quality, fruit harvesting/picking, and adequate transport. In addition, biological problems are also part of an agricultural farm [4] (as shown in Figure 1). For example, plant diseases and weeds are serious threats to the growth of agricultural products and a source of economic losses.

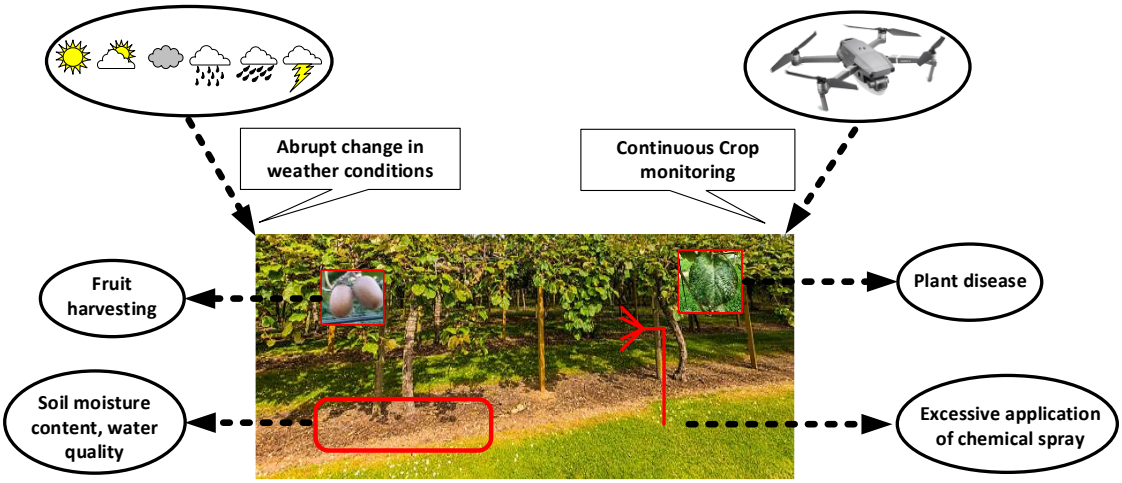


Figure 1. An overview of various factors on agricultural field.

Figure 2 shows a Venn diagram depicting the three major causes/factors of disease in plants. The first factor is the susceptible host, which comprises the existence of a pathogen on the particular host plant, lack of resistance to the pathogen, and incidence of the pathogen at the correct growth stage. The next

is the favorable environment in terms of temperature, lighting conditions, humidity, and other characteristics. The last factor is the correct pathogen that has the ability to cause a disease in plants.

Similarly, the presence of weeds is another problem as they are the costliest agricultural pests and are considered unwanted plants. Weeds are usually deprive the crops of water, sunlight, nutrition, soil moisture, and growing space [5]. Therefore, weeds affect crop productivity and reduce irrigation efficiency. Sometimes, weeds are also a source of shelter for plant diseases and pests. Moreover, when weed seeds are harvested with the crop, they degrade the overall quality of the crops and reduce the market price of the produce.

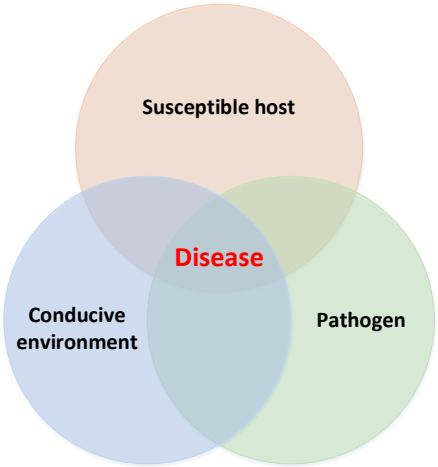


Figure 2. Venn diagram for causes of plant diseases.

To prevent low crop yield due to the occurrence of diseases and weeds in plants, the main agricultural operations are their correct detection, reducing their spread to neighboring plants, and implementation of control treatments. For the identification of plant diseases and weeds, a visual investigation by plant pathologists and biological reviews are generally performed [6]. However, these techniques are typically time-consuming, less accurate, and cost ineffective. Additionally, these conventional methods increase the chance of wrong detection of bacteria, viruses, and weeds in plants. This also leads to adverse environmental conditions due to the excessive application of fungicide/herbicide spray. Therefore, the precise identification of plant diseases and weeds is one of the most essential steps in improving crop productivity/quality, helping growers apply appropriate crop protection remedies, and making smart decisions for future growth.

1.2. Research motivation

The requirement of high food demand can be addressed by maximizing the production of agricultural crops. In this regard, the agricultural problems like the presence of plant diseases and weeds are the source of loss in crop yield and degrade the quality of the crop. Hence, the accurate identification of

the disease and weeds in plants is required to enhance the crop productivity and contribute to meeting global food necessity.

Plant diseases and weeds are controlled chemically by using a fungicide/herbicide spray. Their unnecessary application could increase the global emission of greenhouse gases (GHG) from agriculture. New Zealand's current GHG emission is 30% which has been planned/targeted to net-zero by 2050 (except for the biogenic methane) [7]. One of the elements in achieving this aim is the precise detection of plant diseases and weeds. The successful outcomes could reduce the use of a chemical spray on the infected plants, which would eventually be beneficial to reduce the GHG emission. Furthermore, the additional cost involve in the crop protection can be significantly reduced.

1.3. Research questions

The main objective of this thesis is to improve the identification of plant disease and weeds by using, optimizing, and modifying Deep Learning (DL) architectures and methods, considering various challenges in real agricultural fields. This research also aims to answer the following research questions.

- What are the areas of improvement in DL-based plant disease and weed detection?
- How different performance optimization techniques affect the performance of DL models for the recognition of plant diseases and weeds?
- Can DL have the strength of identifying diseases in different organs of plants along with multiple diseases in plant organs at a time?
- Can DL perform the detection of plant disease in various crops with the same trained/optimized model?
- Can a DL-based method correctly distinguish symptomatically identical plant diseases and weeds?
- How well can DL-based approaches be validated on other datasets with similar classes used for research?

1.4. Scope and limitations

1.4.1. Proposed datasets: This thesis presents two newly generated datasets that contain the disease and healthy classes of 13 different horticultural crops. The proposed datasets superior as compared to the previous studies that majorly relied on disease on single plant organ and considering the detection of plant disease on only one crop with the same trained DL model. However, the limitation is that the images for plant disease classes of various crops like grapevine, avocado, kiwifruit, broccoli, cabbage, kumara, and tomato, are only taken for one organ. Another limitation is that the

weed detection has been performed on an open-source dataset called DeepWeeds, collected in northern Australia. Therefore, no dataset was generated in the New Zealand agricultural environment.

1.4.2. Optimize the deep learning-based object detection models: This thesis proposes various deep learning-based methodologies to improve the mean average precision (mAP) of plant disease and weed identification with a substantial margin. This objective has been achieved by optimizing/using single-stage and two-stage neural networks including Single-shot MultiBox Detector (SSD) [8], Faster Region-based Convolutional Neural Network (R-CNN) [9], Region-based Fully Convolutional Networks (R-FCN) [10], You look Only Once (YOLO-v4) [11, 12], RetinaNet [13], EfficientDet [14], and CenterNet [15]. The main reason for relying only on these types of models is that there are noticeable research gaps in performing plant disease and weed detection using DL-based object detectors. For example, a comprehensive analysis of the effects of various performance optimization techniques on the DL model has not yet been performed. Furthermore, the evaluation of the DL model in terms of the anchor box specifications has not been focused on yet. Moreover, the transfer learning approach has not been evaluated using the optimized weights through different agricultural datasets. Therefore, the selected agricultural tasks are targeted throughout this thesis, using detection-based DL models. In addition, all research has been focused on the classification and detection of plant diseases and weeds. The limitation is that the semantic segmentation-based DL models were not explored for this thesis.

1.4.3. Propose deep learning-based approaches: The current literature related to the deep learning-based detection of plant disease and weeds mainly consists of the modification of the classification/detection models in the hidden layers of the neural network. However, the presented methodologies in this research highlight the earlier steps to get solid grounds for proposing any modifications in the architecture and layers of the well-known DL models.

1.4.4. Deep learning optimizers: This thesis addresses the importance of deep learning optimization algorithms. Most of the work has been done on DL optimizers such as stochastic gradient descent (SGD) with momentum, root mean square propagation (RMSProp), and adaptive moment estimation (Adam). Detailed analyses have been presented for plant disease and weed detection tasks using these three DL optimizers. However, some other methods, such as Adadelta, Adamax, and Adagrad, are used for the classification of plant diseases, as presented in Chapter 3. This limitation in the thesis is due to the reason that all studies related to the detection of plant disease and weeds were performed using TensorFlow Object Detection API and the YOLO/DarkNet framework. These frameworks contain only three DL optimizers, such as SGD with momentum, RMSProp, and Adam. Therefore, advanced DL optimization methods such as Nadam, Ranger, and many others could not have been used.

1.4.5. Comparative analysis of classification models: The comprehensive evaluation of the deep learning models have been performed for the plant disease classification, presented in Chapter 3. The main aim of that part of the research was to understand the behavior of the classification models used for the feature extraction in the DL-based object detectors. However, this kind of analysis has not been performed for weed detection but could be considered in future studies.

1.4.6. Effectiveness of the proposed approaches: This research has shown the effectiveness and robustness of the proposed methodologies through various performance metrics such as mean Average Precision (mAP), training/validation loss, F1-score, precision, recall, and accuracy. Starting with the classification of plant diseases, the proposed approach achieved improved performance compared to the existing techniques in the literature. The usefulness of the proposed methods for tasks like plant disease and weed detection (which contain both classification and localization) has been shown by comparing with the default settings of the best-obtained DL models and other DL architectures.

The difference in proving the novelty and effectiveness of the proposed approaches for particular agricultural tasks is due to a few reasons. First, the detection of plant disease and weed tasks presented in this thesis has been performed for the first time on PlantVillage (with all 38 classes), NZDLPlantDisease-v1&v2 (newly generated plant disease datasets), and DeepWeeds (for the weed detection operation). Therefore, there are no available models to compare the performance with the optimized/modified models presented in this thesis. Furthermore, it was noticed that the modified versions of the DL model in the former studies for plant disease and weed identification/detection had certain logical explanations based on different datasets and classes. On the other hand, various recent studies have been conducted on the same PlantVillage dataset for the particular application of plant disease classification. Hence, the performance of the final model (presented in Chapter 3) has been compared with several DL models found in the literature. In contrast to previous studies, the robustness of the proposed DL approaches/models for both agricultural tasks have been shown on an external testing dataset (generated by random Internet search) and the stratified k-fold cross-validation method.

1.5. Research Contributions

Based on the research questions stated in the previous section, this thesis provides various research ideas and methodologies to improve the performance of deep learning models. This section identifies contributions to the agricultural field of research by deep learning.

1.5.1. Comparative analysis of deep learning models and optimization algorithms

The research work was started by breaking the task of plant disease identification into two parts including plant disease classification (classify disease in several plants) and detection (both classify and localize disease spots in various plants).

State-of-the-art DL meta-architectures contain classification models used as feature extractors and novel architectural design/characteristics for localizing certain objects. Therefore, first, a relatively easier task of classifying plant diseases was performed using 18 classification models. At the time of this research work, none of the articles presented such a comprehensive analysis of DL models that contained different versions of well-known, modified, and cascaded DL architectures. Later, DL meta-architectures were used and trained on the same dataset called PlantVillage. This has been done for the first time performing the plant disease detection in all 38 healthy/disease classes. It was also observed from the literature that DL optimizers have not been used to improve the performance of plant disease detection and classification operations. The successful outcomes of the comparative analysis of DL models and optimizers for these agricultural tasks encouraged the use of this methodology in the rest of the studies.

1.5.2. A deep learning-based weed detection pipeline

The next phase of the research emphasized the identification of weeds. This agricultural operation lacked empirical/experimental evidence before modifying state-of-the-art DL architectures to improve the detection of weed classes. To address this research gap, various aspects of DL were analyzed. The effects of image resizing techniques, image interpolators, weight initialization, batch normalization, and DL optimization algorithms were studied. This research provided insight into DL-based weed detection and provided a strong motivation for the next phase of the research.

1.5.3. Enhanced anchor box approach for weed detection

Most previous studies focused on the modification of state-of-the-art DL classification models/feature extractors for weed identification. However, a detailed architectural evaluation of the DL meta-architecture was absent. Hence, this stage of the research was focused on the major novelty of the best DL model obtained in the previous stage of the study (Faster R-CNN ResNet-101). It was empirically found that the generation of the anchor box has a significant impact on the weed detection results. Therefore, two parameters, such as the anchor box scale and aspect ratio, were modified. For adjusting the scale size, gradual enhancements were proposed, while the aspect ratios were enhanced both reciprocally and gradually. The robustness of the method has been shown by the stratified k-fold cross-validation method and testing on an external testing dataset.

1.5.4. NZDLPlantDisease-v1 & v2: Plant disease datasets for economically important horticultural crops of New Zealand

Important research gaps in open-source datasets of plant diseases and complex problems in horticultural fields have been addressed in this thesis. In this regard, new plant disease datasets are generated, named NZDLPlantDisease-v1 & v2, from various horticultural fields of Auckland and Palmerston North. The proposed datasets have a great significance in terms of addressing the research gaps such as plant diseases in different crops, the presence of multiple diseases in plant organs at a time, and diseases in multiple organs. Moreover, natural field and environmental variations were considered while collecting the dataset images, such as the absence/presence of shadow and different illumination conditions. These datasets are among the prominent contributions of this thesis. The first version consists of the economically important horticultural crops of New Zealand that generate large export values, including apple, avocado, grapevine, kiwifruit, and pear. The second version of the proposed dataset contains diseases in New Zealand's prominent vegetables such as beans, broccoli, cabbage, cauliflower, kumara, pea, potato, and tomato.

1.5.5. A performance-optimized deep learning-based detection of plant disease

To further extend the effectiveness of the presented framework/pipeline for weed detection, similar DL-based approaches have been adopted for plant disease identification. The NZDLPlantDisease-v1 dataset was used that contains 20 healthy/disease classes for different organs of the crops. The ability of DL was investigated to address various research gaps for the first time, including the detection of disease in several organs of plants such as leaves, stems, and fruits, the identification of disease in different crops, and the recognition of multiple diseases in plant organs, with the same trained/optimized model. Some common research flaws were taken into account by this research. For instance, the data augmentation was performed after splitting the dataset into training, validation, and testing sub-datasets, rather than applying the augmentation methods before dividing the dataset. This was to avoid the biased distribution of the dataset. Some additional steps were taken to solve various research gaps in the literature. For example, a comprehensive analysis of the best-suited DL model named region-based fully convolution networks (R-FCN) was performed by evaluating position-sensitive score maps along with anchor box specifications. Furthermore, the novelty/contributions of this research were shown using two techniques, including the stratified five-fold cross-validation technique and testing the optimized/modified DL model on an externally generated dataset. This was done to ensure the robustness of the proposed methodology for the same plant diseases in different agricultural conditions. This study presented new insights into the identification of DL-based plant diseases, compared to other redundant studies.

1.5.6. A transfer learning approach by weight optimization on the agricultural datasets for plant disease detection

The last study in this thesis used the second part of the proposed dataset that contains 28 healthy and disease classes of eight important New Zealand vegetables named NZDLPlantDisease-v2. The proposed approach was based on transfer learning and its importance has been revealed more comprehensively. The main idea consists of using the weights obtained by the general purpose dataset (Common Objects in Context (COCO)) in the first stage for finding the best suited DL architecture. Then, the weights were acquired through three different agricultural datasets including PlantVillage, NZDLPlantDisease-v1, and DeepWeeds in the later stage. However, the updated weights were obtained by using the weed detection pipeline presented in Chapter 5. Finally, the new/updated weights were used to train the best DL model on the proposed dataset. Furthermore, the mean average precision of the final model was compared with the model having optimized weight through the large/general-purpose dataset. This methodology significantly improved the performance of the Faster RCNN Inception ResNet-v2, the best-obtained DL model.

1.5.7. Summary of the deep learning-based approaches in this thesis

A broader summary of the research approaches in the literature and the deep learning-based approaches presented in this thesis for the detection of plant diseases and weeds is shown in Figure 3.

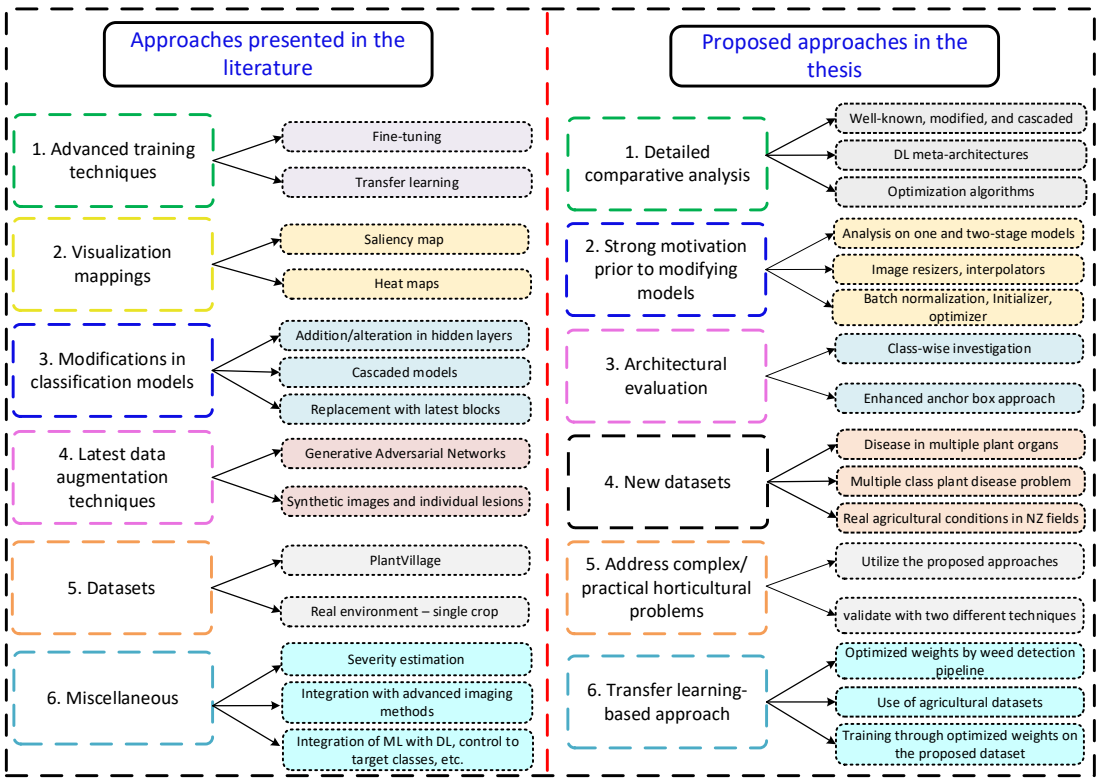


Figure 3. A summary of deep learning-based approaches in the literature and the presented approaches in this thesis.

1.6. An Overview of Deep learning

When the scientific community started to focus on agricultural tasks with Artificial Intelligence (AI)-based techniques, various conventional Machine Learning (ML) algorithms were explored [16]. However, the advent of a subset of ML called Deep Learning (DL) revolutionized object detection research and has proven to have great potential in terms of high accuracy [17]. The DL models showed their capability for the extraction of different patterns/features and their adaptability to certain changes in the characteristics of various objects. Consequently, the recent literature has revealed the effectiveness of the DL architectures over traditional ML models for different agricultural operations including fruit recognition/counting, plant identification, agricultural land cover classification, plant disease detection, and weed identification [3].

The classification and localization of diseases and weeds in plants by deep learning have been an active research topic. Based on the latest advances in computer vision and object identification, DL was chosen as the brain of this Ph.D. for the identification of plant diseases and weeds. Several types of disease were considered in different organs of several plant species. Similarly, numerous types of weeds were targeted for this research.

1.6.1. Basic concepts of deep learning

Deep learning was inspired by the functions and architecture of the human brain. It contains a neural network that has a series of hidden layers to extract the distinct features of the target classes/objects. There are several types of deep neural networks such as MultiLayer Perceptron (MLP), Convolution Neural Network (CNN) [18], Recurrent Neural Network (RNN), Long Short-Term Memory networks (LSTM) [19], Generative Adversarial Network (GAN) [20], Deep Belief Networks (DBN) [21] and AutoEncoders (AE) [22].

After the introduction of AlexNet in 2012 [18], CNN has achieved remarkable progress in image classification tasks and has received significant encouragement from the scientific community. This is due to the automatic feature extraction capability of CNN, which requires a structure of many convolutional layers to classify and detect several objects in a single architecture. In Figure 4, a general representation of CNN is provided for an image classification task. A CNN consists of an input image, hidden layers which are the combination of convolutional and pooling layers, then a fully connected layer is placed, and finally, the output layers are obtained by the SoftMax function. The basic concepts and significance of these layers are summarized in the following.

a. **Convolutional layer:** This layer applies filters to extract the specific features of the images. It requires a certain number of filters along with the size of the filter, specific activation functions

like rectified linear activation function (ReLU), exponential linear unit (ELU), Swish, etc., and the number of strides.

b. Pooling layer: This layer works for the dimensionality reduction of the neural network and requires the size of the window and the number of strides. It could be of two types, such as the max-pooling layer that takes the maximum value in a filter or the average pooling layer that takes the average value in a filter.

c. Fully connected and output layers: The fully connected layer is placed before the output, flattening the results just before the classification of objects, and taking aggregation of feature maps. It requires nodes and activation functions such as ReLU. Then the final classification results are taken by using the SoftMax function, which requires the number of classes in a certain task.

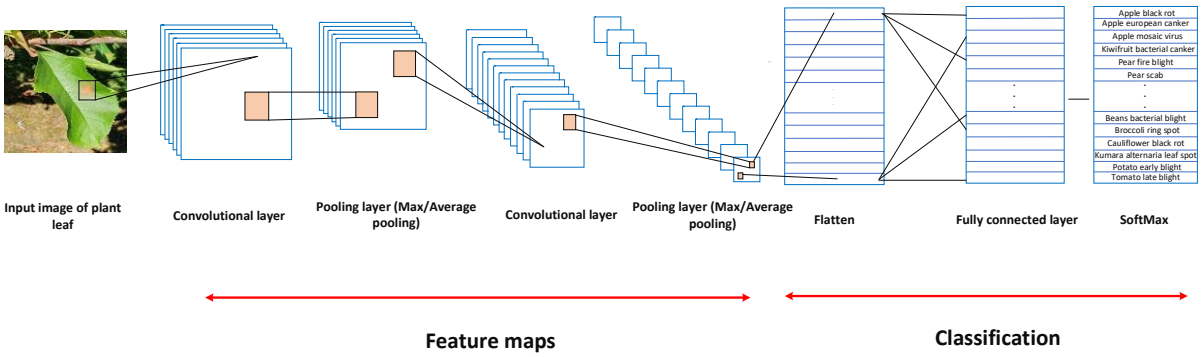


Figure 4. A general representation of a Convolutional Neural Network architecture.

The concept of a deeper network came from the structure of various hidden layers connected to obtain classification results in which the output of the previous layer acts as an input to the next layer. For example, a neural network has an input image that applies to the convolutional layer that further connects to other layers like pooling layers, so the output obtained by the convolutional layer is considered as input to the pooling layer, and so on. Each layer has a certain significance in the classification of objects as the starting hidden layers extracts the basic features of an image such as corners and edges. The next/middle layers detect specific parts of the object like spots of disease in plants. Finally, the last layers detect the overall objects that train with various positions, brightness, contrast, sharpness, and other data augmentation techniques.

1.6.2. Deep learning optimizers

In this thesis, several DL optimizers were used to improve the performance of DL models. The basic characteristics of these optimization algorithms are given as follows:

a. Stochastic Gradient Descent (SGD) with momentum: SGD is one of the most commonly used deep learning optimizers. It has a static learning rate for all parameters required for training the

neural network. The momentum version of SGD has a fast convergence ability [23]. The idea is to evaluate the exponentially weighted average of the gradients and use the gradient to update the weights and biases of the neural network.

b. Adagrad: This optimizer uses different learning rates for every parameter in the neural network. It updates the learning rate according to the occurrence of the update of each parameter [24].

c. Root Mean Square Propagation (RMSProp): RMSProp optimizing algorithm has been proposed to reduce the training time perceived in Adagrad. It works on the idea of using the moving average of the squared gradient and dividing the gradient by the square root of the mean square [25]. The RMSProp restricts the oscillations that produce during loss optimization in the direction of bias, leading to selecting a large learning rate.

d. Adadelta: Adadelta is an extended version of the Adagrad optimizer and assembles the previous gradients over a fixed time window to ensure the continuance of learning after several iterations [26].

e. Adaptive Moment Estimation (Adam): Adam has the combined advantages of Adagrad and RMSProp optimizers. It contains the estimations of the first (mean) and second moments (un-centered variance) of the gradient that are used to adapt the learning rate for each weight of the neural network [27]. Adam takes squared gradients to scale the learning rate and uses the moving average of the gradients.

f. Adamax: A version of Adam has been also proposed named Adamax which is based on the infinity norm. It is useful for the updates of sparse parameter like word embedding. In this method, the weights of the neural network use the maximum of previous weights and the current absolute value of gradients [27].

1.6.3. Performance metrics

Throughout this thesis, the DL models are evaluated by various performance metrics, described as under:

a. Accuracy: Accuracy defines the proportion of correct predictions among overall predictions.

$$Accuracy = \frac{True\ positive + True\ negative}{True\ positive + True\ negative + False\ positive + False\ negative} \quad (1)$$

True positive is the correctly predicted outcome of a positive class; true negative is the correctly predicted outcome of a negative class; false positive is the wrong prediction of a positive class; false negative is the wrong prediction of a negative class

b. Precision: Precision tells the proportion of positive detections/identifications that are actually correct.

$$Precision = \frac{True\ positive}{True\ positive + False\ positive} \quad (2)$$

c. Recall: Recall defines the proportion of actual positives that are identified correctly.

$$Recall = \frac{True\ positive}{True\ positive + False\ negative} \quad (3)$$

d. F1 score: The F1 score is the harmonic mean of precision and recall.

$$F1\ score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (4)$$

e. Mean Average Precision: Mean Average Precision (mAP) is the average of all individual Average Precisions (APs). Whereas, an AP is evaluated across all unique recall levels. At a certain recall level *r*, the interpolated precision (p_{int}) is specified as the highest precision for a recall level $r' \geq r$.

$$p_{int}(r) = maxp(r)' \quad (5)$$

Where *p(r)'* is the measured precision at a maximum recall *r'*.

Then, 11-point interpolated formula is applied to evaluate the mean precision at 11 recall rates equally spaced from [0, 0.1, 0.2, … 1.0].

$$AP = \frac{1}{11} \sum_{r\varepsilon\{0,0.1,0,2,\ldots,1\}}^{n} p_{int}(r) \quad (6)$$

$$mAP = \frac{\sum_{i=1}^{n} AP_i}{n} \quad (7)$$

1.6.4. Evolution in CNNs

The CNNs require a large amount of data to extract the features of the objects. Therefore, at the early age of DL, CNNs take a long training time to achieve model convergence. This problem was solved with the use of graphical processing units (GPUs). Similarly, various developments have been observed in the last decade, in terms of advanced training techniques like transfer learning/fine-tuning [28-30]. Furthermore, several DL-based techniques have been presented for data augmentation, such as generative adversarial networks (GANs) [31]. Several new concepts of neural network layers were also evolved, for example, inception module [32], residual connections [33], depthwise separable convolution [34], residual attention network [35], and convolutional block attention [36]. These advancements have encouraged researchers to explore various real-life problems/applications, including medical image analysis [37], self-driving cars [38], text recognition [39, 40], financial trading [41], and numerous others, using deep learning.

1.6.5. General workflow of deep learning-based plant disease and weed detection

In the last few years, deep learning has produced successful outcomes for plant disease detection, with the major/prominent work done on the PlantVillage dataset. This has opened a vast range of experimentation to get insight into various practical/real agricultural field problems. Similarly, recent studies have also focused on the detection of weeds using various state-of-the-art DL models. A general workflow is presented in Figure 5 showing the steps required to perform the deep learning-based plant disease and weed detection tasks.

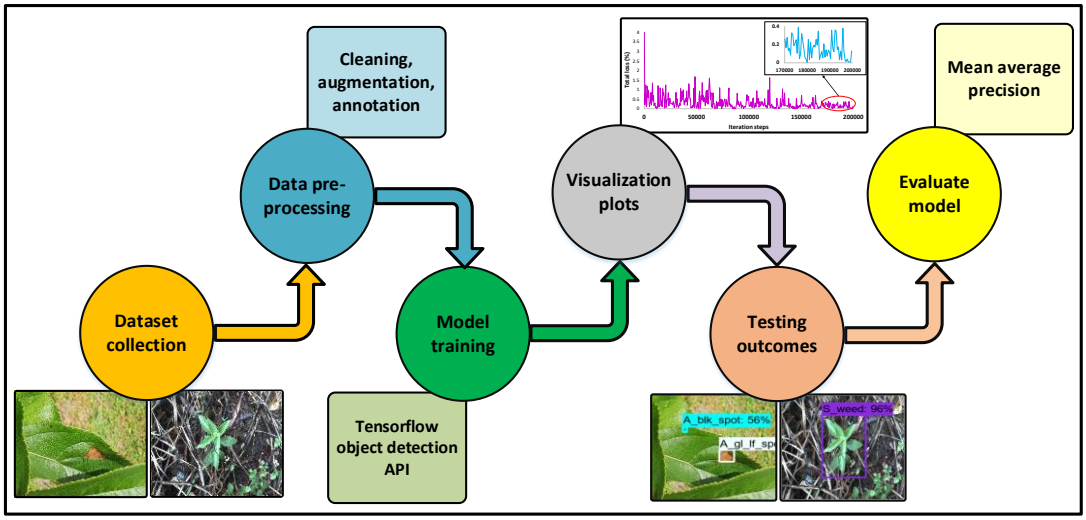


Figure 5. A general workflow for deep learning-based plant disease and weed detection.

1.7. Thesis layout

The chapters in this thesis are structured as follows.

Chapter 2 consists of a comprehensive literature review of deep learning-based plant disease detection. The review starts with a general flow diagram for recognizing plant disease by deep learning. Several visualization techniques that have been used to spot the symptoms of plant disease are summarized. Similarly, the significance of hyperspectral/multispectral imaging with DL architectures is presented. A summary is also provided containing valuable information such as the most used DL models with their performance on different datasets and selected crops in recent studies. The review also outlines the research gaps to get more interesting/clearer outcomes. The literature survey has been published in a Q1 journal named MDPI Plants and received the 2021 best paper award.

Chapter 3 contains a comprehensive comparative analysis of 18 Convolution Neural Networks (CNN) divided into three categories: well-known, modified, and cascaded models. The performance of the best-selected models was optimized by training through various deep learning optimizers. All

experiments were carried out on the PlantVillage dataset, which contains 26 different diseases belonging to 14 plant species. This chapter has been published in MDPI Plants.

In **Chapter 4**, the performance of three DL meta-architectures was evaluated, including various versions of the Single Shot MultiBox Detector (SSD), Faster Region-Based Convolutional Neural Network (R-CNN), and Region-Based Fully Convolutional Networks (R-FCN). Moreover, state-of-the-art deep learning optimizers were attempted to improve the mean average precision (mAP) of the best-obtained deep learning architecture. This chapter has also been published in MDPI Plants.

Chapter 5 elaborates a DL-based pipeline for the identification of eight classes of weeds and a negative class. The proposed methodology consists of obtaining the best-suitable model divided into single-stage and two-stage neural networks. Next, the effects of image resizing techniques (aspect ratio and fixed shape) were studied along with four image interpolation methods (bilinear, bicubic, area, and nearest neighbor). It led to optimization of the weights of the best-acquired model using initialization techniques such as truncated normal, scaling variance, and random normal. Then batch normalization was applied, and its effects were evaluated. Finally, DL optimization algorithms like SGD with momentum, RMSProp, and Adam were used. The proposed approach significantly improved the performance of the DL model compared to the default settings. All results were taken from the DeepWeeds dataset. The results were validated by the stratified k-fold cross-validation technique. Another contribution from this work is that all important data have been made publicly available online, such as the annotated DeepWeeds dataset, configuration files for each experiment, the inference graph of the final optimized DL model, and dataset folds for validation purposes. This work has been published in a Q1 journal named Frontiers in Plant Science.

The work undertaken in **Chapter 6** is the continuation of the previous chapter. A strong research motivation was derived to further improve the performance of the deep learning model. This phase of the research was dedicated to improving the average precision (AP) of the chinee apple that attained a low AP in the previous phase of the research. Moreover, maintaining the AP of other classes of weeds and a negative/non-weed class was also aimed. In this regard, an enhanced anchor box approach has been presented that consists of empirical modification/enhancement of anchor box scales and aspect ratios of the Faster R-CNN model. The robustness of this research was shown by the stratified k-fold cross-validation method and testing on an externally generated dataset. This work has been published in a Q1 journal named Agronomy.

In **Chapter 7**, we presented a new plant disease dataset called NZDLPlantDisease-v1. It contains 20 healthy and disease classes from five of the most important horticultural crops in New Zealand,

including apple, avocado, grapevine, kiwifruit, and pear. This research utilized various steps taken for weed identification, as presented in the previous chapter, with an additional two steps. For example, data augmentation techniques were divided into five categories, and empirical observations were made on the performance of the model and analyzed position-sensitive score maps. The results of this study have also been validated by two techniques including a five-fold stratified cross-validation method and testing on an external dataset in different agricultural environments. This chapter has been published in a Q1 journal named IEEE Access.

Chapter 8 demonstrates a transfer learning approach through weight optimization on the agricultural datasets. In this regard, the second part in the series of the proposed dataset was used, named NZDLPlantDisease-v2. First, the best-suited DL architecture was obtained by using the pre-trained weights on a large general-purpose dataset (COCO). Later, the best DL model was trained in the presence of data augmentation methods. The model was then trained and optimized on three agricultural datasets; two of them were the plant disease dataset (PlantVillage and NZDLPlantDisease-v1) and a weeds dataset (DeepWeeds). The weight optimization was achieved by the use of image resizer, interpolators, weight initializers, batch normalization, and DL optimizers. Finally, the updated/new weights from all three agricultural datasets were used to retrain the DL model on the proposed dataset. The performance was significantly improved and validated by cross-validation and an external dataset. This work has been published in a Q1 journal named Frontiers in Plant Science.

Finally, **Chapter 9** includes the overall conclusion of the thesis along with various directions for future work to further advance research on deep learning-based plant diseases and weed identification.

Appendix 1 presents a literature survey of various machine learning (ML) and deep learning (DL) models for the five selected agricultural applications, including detection and classification of plant diseases/pests, recognition of plants/leaves, discrimination, and classification of crop weeds, harvesting of fruits and vegetables, and classification of agricultural land cover. The focus of the survey was towards the use and implementation of the robotic/automated systems for the selected agricultural applications. The effectiveness of DL over traditional ML models for all agricultural operations was shown, summarizing the results presented in previous studies. Moreover, various important research gaps are highlighted from the previous articles and groundbreaking future directions are noted to achieve further advancement in agricultural automation. This literature survey has been published in a Q1 journal named Precision Agriculture.

1.8. References

[1] FAO, "How to Feed the World 2050: Global agriculture towards 2050," Food and Agriculture of the United Nations (FAO), Rome, 12-13 October 2009. [Online]. Available: https://www.fao.org/fileadmin/templates/wsfs/docs/Issues_papers/HLEF2050_Global_Agriculture.pdf

[2] A. G. Aitken and I. J. Warrington, "Fresh Facts," Plant & Food Research, 2021. [Online]. Available: https://www.freshfacts.co.nz/files/freshfacts-2021.pdf

[3] M. H. Saleem, J. Potgieter, and K. M. Arif, "Automation in agriculture by machine and deep learning techniques: A review of recent developments," *Precision Agriculture*, vol. 22, no. 6, pp. 2053-2091, 2021.

[4] T. N. Liliane and M. S. Charles, "Factors affecting yield of crops," *Agronomy-Climate Change & Food Security; IntechOpen: London, UK*, p. 9, 2020.

[5] A. M. Hasan, F. Sohel, D. Diepeveen, H. Laga, and M. G. Jones, "A survey of deep learning techniques for weed detection from images," *Computers and Electronics in Agriculture*, vol. 184, p. 106067, 2021.

[6] M. H. Saleem, S. Khanchi, J. Potgieter, and K. M. Arif, "Image-based plant disease identification by deep learning meta-architectures," *Plants*, vol. 9, no. 11, p. 1451, 2020.

[7] J. L. Carroll and A. J. Daigneault, "Achieving ambitious climate targets: is it economical for New Zealand to invest in agricultural GHG mitigation?," *Environmental Research Letters*, vol. 14, no. 12, p. 124064, 2019.

[8] W. Liu *et al.*, "Ssd: Single shot multibox detector," in *European conference on computer vision*, 2016: Springer, pp. 21-37.

[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[10] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," *Advances in neural information processing systems*, vol. 29, 2016.

[11] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[12] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[13] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980-2988.

[14] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10781-10790.

[15] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Centernet: Keypoint triplets for object detection," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6569-6578.

[16] K. G. Liakos, P. Busato, D. Moshou, S. Pearson, and D. Bochtis, "Machine learning in agriculture: A review," *Sensors*, vol. 18, no. 8, p. 2674, 2018.

[17] M. H. Saleem, J. Potgieter, and K. M. Arif, "Plant disease detection and classification by deep learning," *Plants*, vol. 8, no. 11, p. 468, 2019.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[19] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, 2020.

[20] I. Goodfellow *et al.*, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[21] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527-1554, 2006.

[22] D. Bank, N. Koenigstein, and R. Giryes, "Autoencoders," *arXiv preprint arXiv:2003.05991*, 2020.

[23] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[24] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of machine learning research*, vol. 12, no. 7, 2011.

[25] G. Hinton, N. Srivastava, and K. Swersky, "Neural networks for machine learning," *Coursera, video lectures*, vol. 264, no. 1, pp. 2146-2153, 2012. [Online]. Available: https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf.

[26] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[28] J. G. A. Barbedo, "Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification," *Computers and electronics in agriculture*, vol. 153, pp. 46-53, 2018.

[29] J. Chen, J. Chen, D. Zhang, Y. Sun, and Y. A. Nanehkaran, "Using deep transfer learning for image-based plant disease identification," *Computers and Electronics in Agriculture*, vol. 173, p. 105393, 2020.

[30] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, "A comparative study of fine-tuning deep learning models for plant disease identification," *Computers and Electronics in Agriculture*, vol. 161, pp. 272-279, 2019.

[31] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[32] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818-2826.

[33] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Thirty-first AAAI conference on artificial intelligence*, 2017.

[34] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251-1258.

[35] F. Wang *et al.*, "Residual attention network for image classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3156-3164.

[36] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3-19.

[37] X. Xie, J. Niu, X. Liu, Z. Chen, S. Tang, and S. Yu, "A survey on incorporating domain knowledge into deep learning for medical image analysis," *Medical Image Analysis*, vol. 69, p. 101985, 2021.

[38] J. Ni, Y. Chen, Y. Chen, J. Zhu, D. Ali, and W. Cao, "A survey on theories and applications for self-driving cars based on deep learning methods," *Applied Sciences*, vol. 10, no. 8, p. 2749, 2020.

[39] Y. Chen, H. Shu, W. Xu, Z. Yang, Z. Hong, and M. Dong, "Transformer text recognition with deep learning algorithm," *Computer Communications*, vol. 178, pp. 153-160, 2021.

[40] C. Zhang, W. Ding, G. Peng, F. Fu, and W. Wang, "Street view text recognition with deep learning for urban scene understanding in intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4727-4743, 2020.

[41] A. M. Ozbayoglu, M. U. Gudelek, and O. B. Sezer, "Deep learning for financial applications: A survey," *Applied Soft Computing*, vol. 93, p. 106384, 2020.

Chapter 2. Plant Disease Detection and Classification by Deep Learning

This chapter contains content from the following article.

M. H. Saleem, J. Potgieter, and K. M. Arif, "Plant disease detection and classification by deep learning," Plants, vol. 8, no. 11, p. 468, 2019. https://doi.org/10.3390/plants8110468

plants

MDPI

Review

Plant Disease Detection and Classification by Deep Learning

Muhammad Hammad Saleem [1], Johan Potgieter [2] and Khalid Mahmood Arif [1,*]

[1] Department of Mechanical and Electrical Engineering, School of Food and Advanced Technology, Massey University, Auckland 0632, New Zealand; h.saleem@massey.ac.nz
[2] Massey Agritech Partnership Research Centre, School of Food and Advanced Technology, Massey University, Palmerston North 4442, New Zealand; j.potgieter@massey.ac.nz
* Correspondence: k.arif@massey.ac.nz

Received: 25 September 2019; Accepted: 29 October 2019; Published: 31 October 2019

Abstract: Plant diseases affect the growth of their respective species, therefore their early identification is very important. Many Machine Learning (ML) models have been employed for the detection and classification of plant diseases but, after the advancements in a subset of ML, that is, Deep Learning (DL), this area of research appears to have great potential in terms of increased accuracy. Many developed/modified DL architectures are implemented along with several visualization techniques to detect and classify the symptoms of plant diseases. Moreover, several performance metrics are used for the evaluation of these architectures/techniques. This review provides a comprehensive explanation of DL models used to visualize various plant diseases. In addition, some research gaps are identified from which to obtain greater transparency for detecting diseases in plants, even before their symptoms appear clearly.

Keywords: plant disease; deep learning; convolutional neural networks (CNN)

1. Introduction

The Deep Learning (DL) approach is a subcategory of Machine Learning (ML), introduced in 1943 [1] when threshold logic was introduced to build a computer model closely resembling the biological pathways of humans. This field of research is still evolving; its evolution can be divided into two time periods-from 1943–2006 and from 2012–until now. During the first phase, several developments like backpropagation [2,3], chain rule [4], Neocognitron [5], hand written text recognition (LeNET architecture) [6], and resolving the training problem [7,8] were observed (as shown in Figure 1). However, in the second phase, state-of-the-art algorithms/architectures were developed for many applications including self-driving cars [9–11], healthcare sector [12–14], text recognition [6,15–17], earthquake predictions [18–20], marketing [21], finance [22,23], and image recognition [24–29]. Among those architectures, AlexNet [30] is considered to be a breakthrough in the field of DL as it won the ImageNet challenge for object recognition known as ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in the year 2012. Soon after, several architectures were introduced to overcome the loopholes observed previously. For the evaluation of these algorithms/architectures, various performance metrics were used. Among these metrics, top-1%/top-5% error [24,26,30,31], precision and recall [25,32–34], F1 score [32,35], training/validation accuracy and loss [34,36], classification accuracy (CA) [37–41] are the most popular. For the implementation of DL models, several steps are required, from the collection of datasets to visualization mappings are explained in Figure 2.

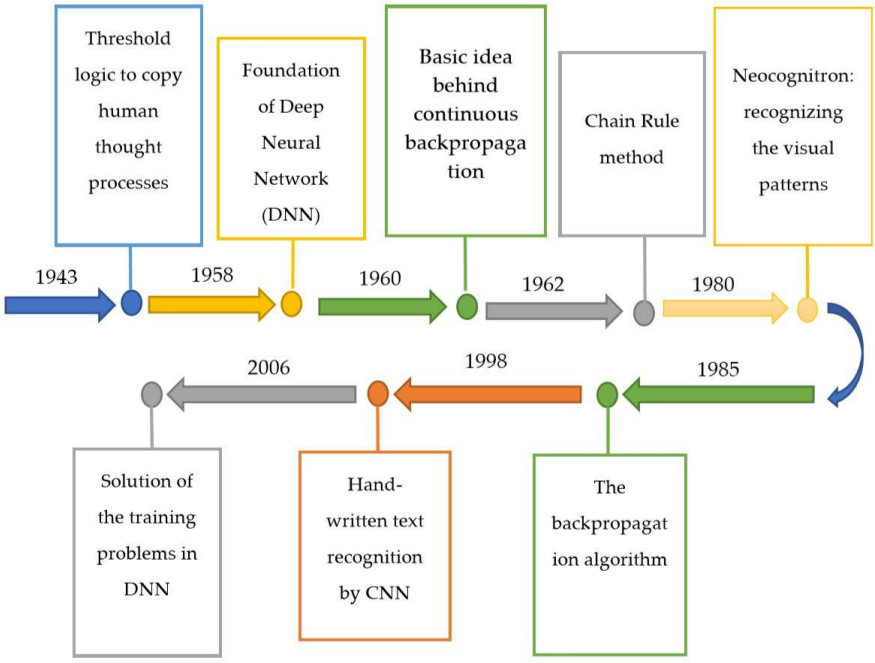


Figure 1. Summary of the evolution of deep learning from 1943–2006.

When DL architectures started to evolve with the passage of time, researchers applied them to image recognition and classification. These architectures have also been implemented for different agricultural applications. For example, in [42], classification of leaves was performed by using author-modified CNN and Random Forest (RF) classifier among 32 species in which the performance was evaluated through CA at 97.3%. On the other hand, it was not as efficient at detecting occluded objects [43]. Leaf and fruit counting were also performed by deep CNN in [44,45] and [46] respectively. For classification of crop type, [47] used author-modified CNN, [36] applied VGG 16, [34] implemented three unit LSTM, and [33] used CNN and RGB histogram technique. [47] used CA, [36] used CA and Intersection over Union (IoU), [34] used CA and F1, and [33] used F1-score as a performance metric. Among them, [33,47] did not provide training/validation accuracy and loss. Moreover, recognition of different plants has been done by the DL approach in [48–50]. [48,50] employed user-modified CNN while [49] used AlexNet architecture. All were evaluated on the basis of CA. [49] outperformed the other two in terms of CA. Similarly, crop/weed discrimination was performed in [51,52], in which the author proposed CNN be used, and two datasets were utilized for the evaluation of the model. [51] evaluated precision and recall; however, [52] obtained CA for the validation of the proposed models respectively. The identification of plants by the DL approach was studied and achieved a success rate of 91.78% [53]. On top of that, DL approaches are also used for critical tasks like plant disease detection and classification, which is the main focus of this review. There are some research papers previously presented to summarize the research based on agriculture (including plant disease recognition) by DL [43,54], but they lacked some of the recent developments in terms of visualization techniques implemented along with the DL and modified/cascaded version of famous DL models, which were used for plant disease identification. Moreover, this review also provides the research gaps in order to get a clearer/more transparent vision of symptoms observed due to diseases in the plants.

The remaining part of the paper is comprised of Section 2, describing the famous and new/modified DL architectures along with visualization mapping/techniques used for plant disease detection; Section 3, elaborating upon the Hyperspectral Imaging with DL models; and finally, Section 4, concluding the review and providing future recommendations for achieving more advancements in the visualization, detection, and classification of plants' diseases.

Plants **2019**, *8*, 468

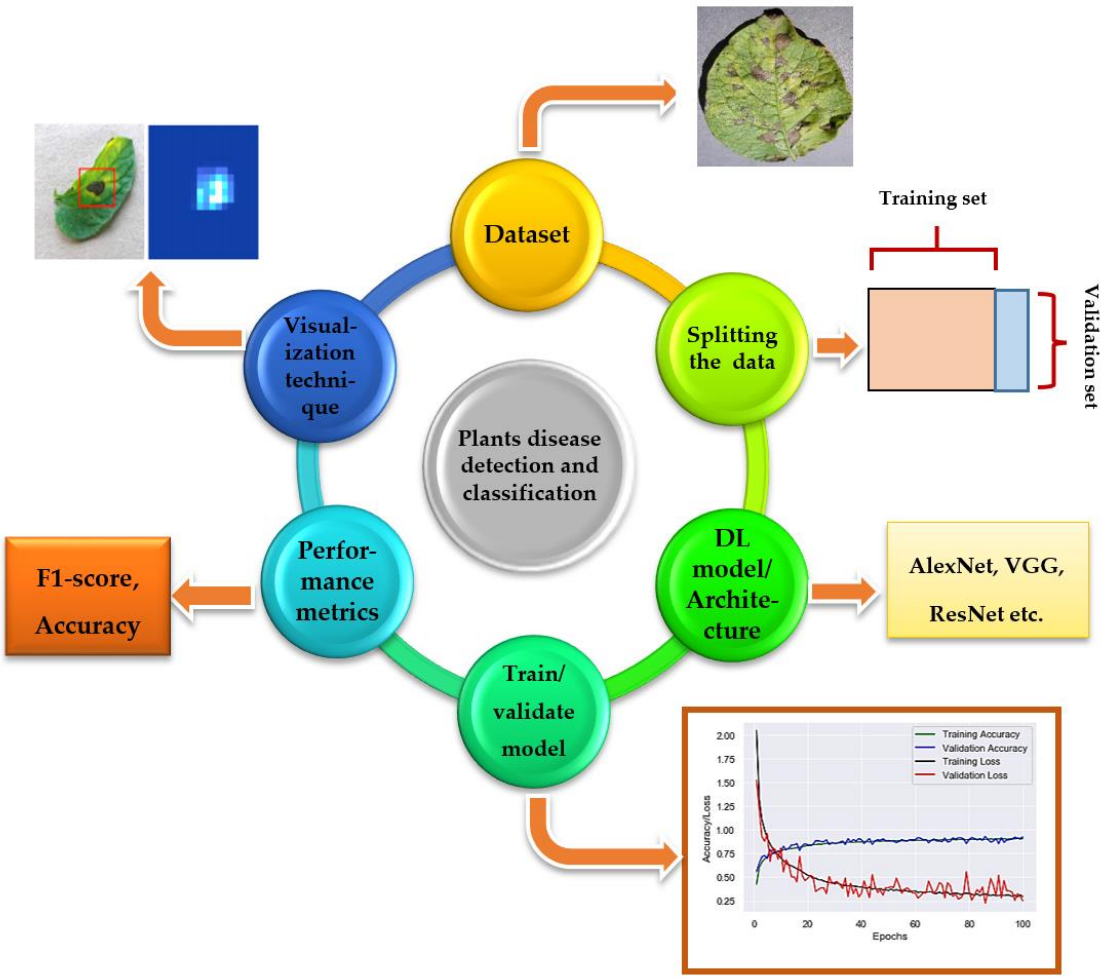


Figure 2. Flow diagram of DL implementation: First, the dataset is collected [25] then split into two parts, normally into 80% of training and 20% of validation set. After that, DL models are trained from scratch or by using transfer learning technique, and their training/validation plots are obtained to indicate the significance of the models. Then, performance metrics are used for the classification of images (type of particular plant disease), and finally, visualization techniques/mappings [55] are used to detect/localize/classify the images.

2. Plant Disease Detection by Well-Known DL Architectures

Many state-of-the-art DL models/architectures evolved after the introduction of AlexNet [30] (as shown in Figure 3 and Table 1) for image detection, segmentation, and classification. This section presents the researches done by using famous DL architectures for the identification and classification of plants' diseases. Moreover, there are some related works in which new visualization techniques and modified/improved versions of DL architectures were introduced to achieve better results. Among all of them, the PlantVillage dataset has been used widely as it contains 54,306 images of 14 different crops having 26 plant diseases [25]. Moreover, they used several performance metrics to evaluate the selected DL models, which are described as below.

Plants **2019**, *8*, 468

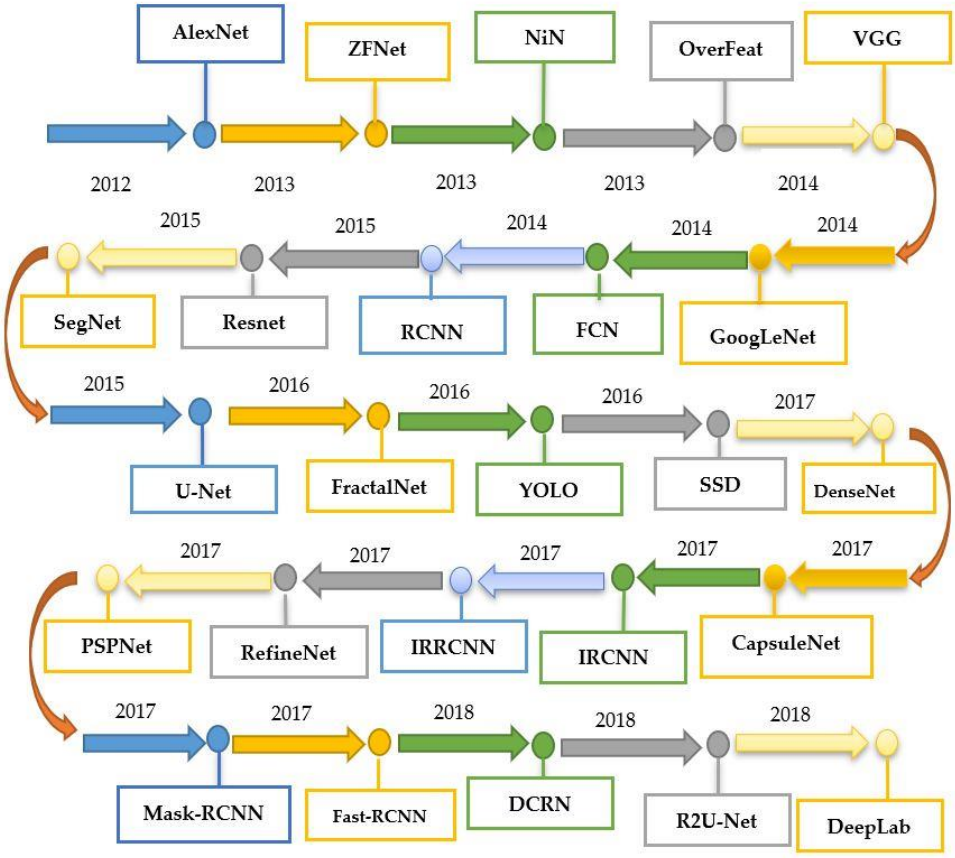


Figure 3. Summary of the evolution of various deep learning models from 2012 until now.

2.1. Implementation of DL Models

2.1.1. Without Visualization Technique

In [56], CNN was used for the classification of diseases in maize plants and histogram techniques to show the significance of the model. In [57], basic CNN architectures like AlexNet, GoogLeNet and ResNet were implemented for identifying the tomato leaf diseases. Training/validation accuracy were plotted to show the performance of the model; ResNet was considered as the best among all the CNN architectures. In order to detect the diseases in banana leaf, LeNet architecture was implemented and CA, F1-score were used for the evaluation of the model in Color and Gray Scale modes [32]. Five CNN architectures were used in [58], namely, AlexNet, AlexNetOWTbn, GoogLeNet, Overfeat, and VGG architectures in which VGG outclassed all the other models. In [35], eight different plant diseases were recognized by three classifiers, Support Vector Machines (SVM), Extreme Learning Machine (ELM), and K-Nearest Neighbor (KNN), used with the state-of-the-art DL models like GoogLeNet, ResNet-50, ResNet-101, Inception-v3, InceptionResNetv2, and SqueezeNet. A comparison was made between those models, and ResNet-50 with SVM classifier got the best results in terms of performance metrics like sensitivity, specificity, and F1-score. According to [59], a new DL model—Inception-v3—was used for the detection of cassava disease. In [60], plant diseases in cucumber were classified by the two basic versions of CNN and got the highest accuracy, equal to 0.823. The traditional plant disease recognition and classification method was replaced by Super-Resolution Convolutional Neural Network (SRCNN) in [61]. For the classification of tomato plant disease, AlexNet and SqueezeNet v1.1 models were used in which AlexNet was found to be the better DL model in terms of accuracy [62]. A comparative analysis was presented in [63] to select the best DL architecture for detection of plant diseases. Moreover in [64], six tomato plant diseases were classified by using AlexNet and VGG-16 DL architectures, and a detailed comparison was provided with the help of classification accuracy. In

the above approaches, no visualization technique was applied to spot the symptoms of diseases in the plants.

Table 1. Comparison of state-of-the-art deep learning models.

Deep Learning Models	Parameters	Key Features and Pros/Cons
LeNet	60k	First CNN model. Few parameters as compared to other CNNmodels. Limited capability of computation
AlexNet	60M	Known as the first modern CNN. Best image recognition performance at its time. Used ReLU to achieve better performance. Dropout technique was used to avoid overfitting
OverFeat	145M	First model used for detection, localization, and classification of objects through a single CNN. Large number of parameters as compared to AlexNet
ZFNet	42.6M	Reduced weights (as compared to AlexNet) by considering 7×7 kernels and improved accuracy
VGG	133M–144M	3×3 receptive fields were considered to include more number of non-linearity functions which made decision function discriminative. Computationally expensive model due to large number of parameters
GoogLeNet	7M	Fewer number of parameters as compared to AlexNet model. Better accuracy at its time
ResNet	25.5M	Vanishing gradient problem was addressed. Better accuracy than VGG and GoogLeNet models
DenseNet	7.1M	Dense connections between the layers. Reduced number of parameters with better accuracy
SqueezeNet	1.25M	Similar accuracy as AlexNet with 50 times lesser parameters. Considered 1×1 filters instead of 3×3 filters. Input channels were decreased. Large activation maps of convolution layers
Xception	22.8M	A depth-wise separable convolution approach. Performed better than VGG, ResNet, and Inception-v3 models
MobileNet	4.2M	Considered the depth-wise separable convolution concept. Reduced parameters significantly. Achieved accuracy near to VGG and GoogLeNet
Modified/Reduced MobileNet	0.5/0.54M	Lesser number of parameters as compared to MobileNet. Similar accuracy as compared to MobileNet
VGG-Inception	132M	A cascaded version of VGG and inception module. The number of parameters were reduced by substituting 5×5 convolution layers with two 3×3 layers. Testing accuracy was increased as compared to many well-known DL models like AlexNet, GoogLeNet, Inception-v3, ResNet, and VGG-16.

2.1.2. With Visualization Techniques

The following approaches employed DL models/architectures and also visualization techniques which were introduced for a clearer understanding of plants' diseases. For example, [55] introduced the saliency map for visualizing the symptoms of plant disease; [27] identified 13 different types of plant disease with the help of CaffeNet CNN architecture, and achieved CA equal to 96.30%, which was better than the previous approach like SVM. Moreover, several filters were used to indicate the disease spots. Similarly, [25] used AlexNet and GoogLeNet CNN architectures by using the publicly available PlantVillage dataset. The performance was evaluated by means of precision (P), recall (R), F1 score, and overall accuracy. The uniqueness of this paper was the implication of three scenarios (color, grayscale, and segmented) for evaluating the performance metrics and comparison of the two

famous CNN architectures. It was concluded that GoogLeNet outperformed AlexNet. Moreover, visualization activation in the first layers clearly showed the spots of diseases. In [65], a modified LeNet model was used to detect olive plant diseases. The segmentation and edges maps were used to spot the diseases in the plants. Detection of four cucumber diseases was done in [66] and accuracy was compared with Random Forest, Support Vector Machines, and AlexNet models. Moreover, the image segmentation method was used to view the symptoms of diseases in the plants. A new DL model was introduced in [67] named teacher/student network and proposed a novel visualization method to identify the spots of plant diseases. DL models with some detectors were implemented in [68], in which the diseases in plants were marked along with their prediction percentage. Three detectors, named Faster-RCNN, RFCN and SSD, were used with the famous architectures like AlexNet, GoogLeNet, VGG, ZFNet, ResNet-50, ResNet-101 and ResNetXt-101 for a comparative study which outlined the best among all the selected architectures. It was concluded that ResNet-50 with the detector R-FCN gave the best results. Furthermore, a kind of bounding box was drawn to identify the particular type of disease in the plants. In [69], a banana leaf disease and pest detection was performed by using three CNN models (ResNet-50, Inception-V2 and MobileNet-V1) with Faster-RCNN and SSD detectors. According to [70], different combinations of CNN were used and presented heat maps as input to the diseased plants' images and provided the probability related to the occurrence of a particular type of disease. Moreover, ROC curve evaluates the performance of the model. Furthermore, feature maps for rice disease were also included in the paper. LeNet model was used in [71] to detect and classify diseases in the soybean plant. In [72], a comparison between AlexNet and GoogLeNet architectures for tomato plant diseases was done, in which GoogLeNet performed better than the AlexNet; also, it proposed occlusion techniques to recognize the regions of diseases. The VGG-FCN and VGG-CNN models were implemented in [73], for the detection of wheat plant diseases and visualization of features in each block. In [74], VGG-CNN model was used for the detection of Fusarium wilt in radish and K-means clustering method was used to show the marks of diseases. A semantic segmentation approach by CNN was proposed in [75] to detect the disease in cucumber. In [76], an approach based on the individual symptoms/spots of diseases in the plants was introduced by using a DL model for detecting plant diseases. A Deep CNN framework was developed for identification, classification, and quantification of eight soybean stresses in [77]. In [78], rice plant diseases were identified by CNN, and feature maps were obtained to identify the patches of diseases. A deep residual neural network was extended in [79] for the development of a mobile application in which a clear identification of diseases in plants was done by the hot spot. An algorithm based on the hot spot technique was also used in [80], in which those spots were extracted by modification in the segmented image to attain color constancy. Furthermore, each obtained hot-spot was described by two descriptors, one was used to evaluate the color information of the disease and other was used to identify the texture of the hot-spots. The cucumber plant diseases were identified in [81] by using the dilation convolutional neural network. A state-of-the-art visualization technique was proposed in [82] by correlation coefficient and DL models like AlexNet and VGG-16 architectures. In [83], color space and various vegetation indices combined with CNN model (LeNet) to detect the diseases in grapes. To summarize, Table 2 outlines some of the visualization mapping/techniques.

Table 2. Visualization mapping/techniques used in several approaches.

Visualization Techniques/Mappings	References
Visualization of features having filter from first to final layer	[27]
Visualize activations in first convolutional layer	[25]
Saliency map visualization	[55]
Classification and localization of diseases by bounding boxes	[68]
Heat maps were used to identify the spots of the disease	[70]
Feature map for the diseased rice plant	[78]
Symptoms visualization method	[72]
Feature and spatial core maps	[73]
Color space into HSV and K-means clustering	[74]
Feature map for spotting the diseases	[77]
Image segmentation method	[66]
Reconstruction of images on discriminant regions, segmentation of images by binary threshold theorem, and heat map construction	[67]
Saliency map visualization	[84]
Saliency map, 2D and 3D contour, mesh graph image	[82]
Activation visualization	[85]
Segmentation map and edge map	[65]

For the practical experimentation of detection of plants' diseases, an actual/real background/environment should be considered in order to evaluate the performance of the DL model more accurately. In most of the above approaches, the selected datasets considered plain backgrounds which are not realistic scenarios for identification and classification of the diseases [25,27,32,56–58,60,61,65,72,77,78], except for a few of them that have considered the original backgrounds [35,59,68,70,73,74]. The output of the visualization techniques used in several researches are shown in Figures 4–11.

In Figure 4, feature maps from the first to the fifth hidden layer are shown as the neuron in a feature map having identical features at different positions of an image. Starting from the first layer (a), the features in feature maps represent separate pixels to normal lines, whereas the fifth layer shows some particular parts of the image (h).

Two types of visualization maps are shown in Figure 5, namely, heat map and saliency map techniques. The heat maps identify the diseases shown as red boxes in the input image, but it should be noted that one disease marked in (d) has not been detected. This problem was resolved in the saliency map technique after the application of the guided back-propagation [55]; all the spots of plant disease were successfully identified thanks to a method which is superior to the heat map.

Plants **2019**, *8*, 468

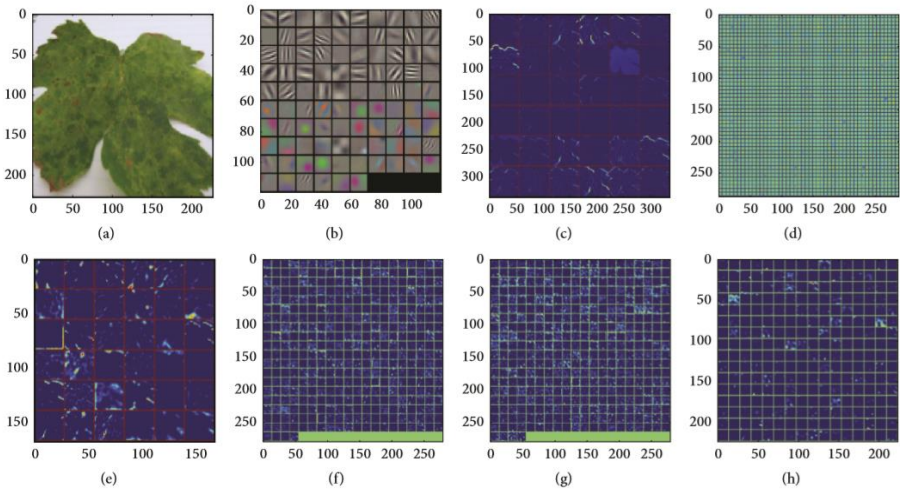


Figure 4. Feature maps after the application of convolution to an image: (**a**) real image, (**b**) first convolutional layer filter, (**c**) rectified output from first layer, (**d**) second convolutional layer filter, (**e**) output from second layer, (**f**) output of third layer, (**g**) output of fourth layer, (**h**) output of fifth layer [27].

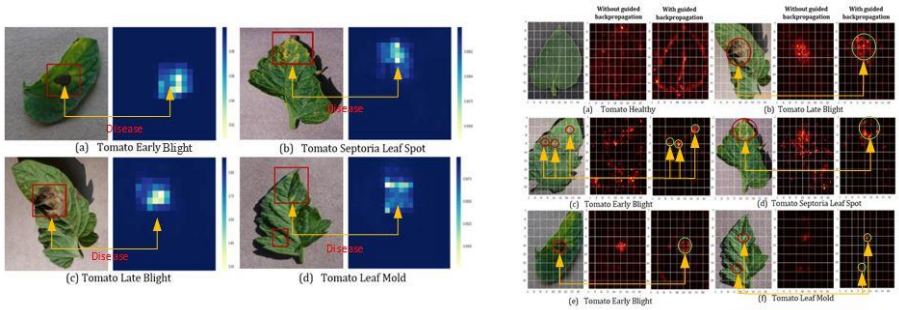


Figure 5. Tomato plant disease detection by heat map: on left hand side (**a**) tomato early blight, (**b**) tomato septoria leaf spot, (**c**) tomato late blight and (**d**) tomato leaf mold) and saliency map; on right hand side (**a**) tomato healthy, (**b**) tomato late blight, (**c**) tomato early blight, (**d**) tomato septoria leaf spot, (**e**) tomato early blight, (**f**) tomato leaf mold) [55].

Figure 6 represents the heat map to detect the disease in maize plants. First, the image was represented in the form of the probability of each portion containing disease. Then, the probabilities were placed into the form of a matrix in order to denote the outcome of all the areas of the input image.

Figure 6. Detection of maize disease (indicated by red circles) by heat map [70].

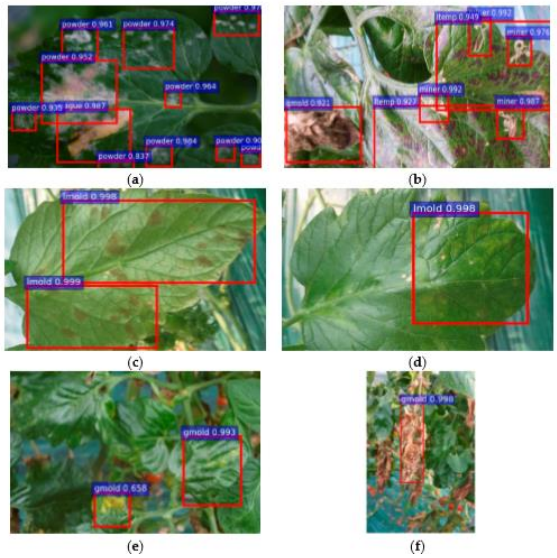


Figure 7. Bounding box indicates the type of diseases along with the probability of their occurrence [68]. A bounding box technique was used in Figure 7 in which (**a**) represents the one type of disease along with its rate of occurrence, (**b**) indicates three types of plant disease (miner, temperature, and gray mold) in a single image, (**c**,**d**) shows one class of disease but contains different patterns on the front and back side of the image, (**e**,**f**) displays different patterns of gray mold in the starting and end stages [68].

A new visualization technique was proposed in [67] as shown in Figures 8 and 9. In Figure 8a, the input image was regenerated for student/teacher architecture [67], and a single channel heat map was produced after the application of simple aggregation on the channels of the regenerated image (Figure 8b). Then, a simple binary threshold algorithm was applied to obtain sharp symptoms of diseases in the plant. Then, [67] indicated the significance of the proposed technique by comparing it with the other visualization techniques as shown in Figure 9. On the left hand side, LRP-Z, LRP-Epsilon, and gradient did not identify plant diseases clearly. However, the Deep Taylor approach produced better results but indicated some portion of the leaf disease. On the right hand side, an imperfect localization of the plant disease was shown in grad-cam techniques which was resolved in the proposed technique by the use of a decoder [67].

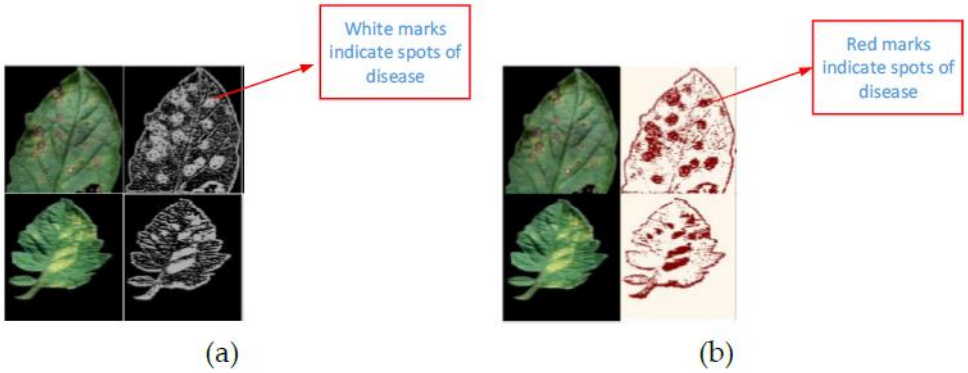


Figure 8. (**a**) Teacher/student architecture approach; (**b**) segmentation using a binary threshold algorithm [67].

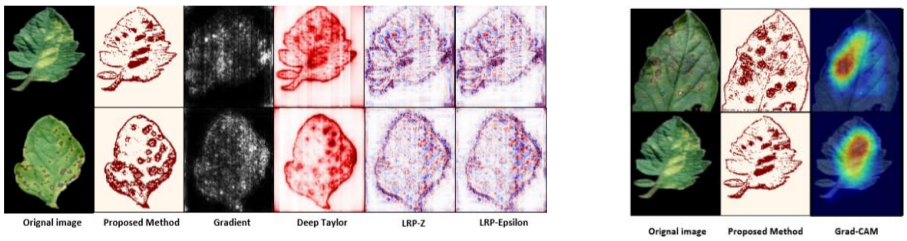


Figure 9. Comparison of Teacher/student approach visualization map with the previous approaches [67].

In order to find the significance of CNN architectures to differentiate between various diseases of plants, the feature maps were obtained as shown in Figure 10. The result proves a good performance of the proposed CNN model as it clearly identifies the disease in plants [85].

In Figure 11 the segmentation and edged maps were obtained to identify the diseases in plants. It is noted that the yellow colored area is marked as white surface in the segmentation map to show the affected part of the leaf.

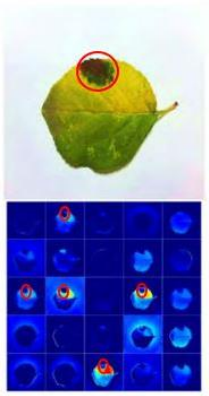


Figure 10. Activation visualization for detection of apple plant disease to show the significance of a VGG-Inception model (the plant disease is indicated by the red circle) [85].

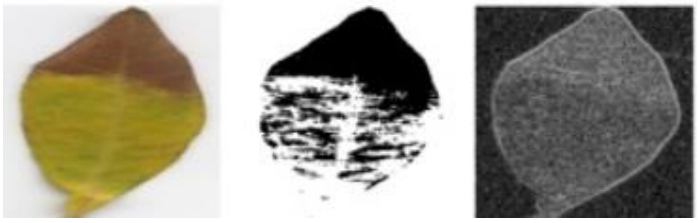


Figure 11. Segmentation and edge map for olive leaf disease detection [65].

2.2. New/Modified DL Architectures for Plant-Disease Detection

According to some of the research papers, new/modified DL architectures have been introduced to obtain better/transparent detection of plant disease, such as [86] presented improved GoogLeNet and Cifar-10 models and their performance compared with AlexNet and VGG. It was found that improved versions of these state-of-the-art models produced a remarkable accuracy of 98.9%. In [87], a new DL model was introduced to obtain more accurate detection of plant diseases as compared to SVM, AlexNet, GoogLeNet, ResNet-20, and VGG-16 models. This model achieved 97.62% accuracy for classifying apple plant diseases. Moreover, the dataset extended in 13 different ways (rotation of 90°, 180°, 270° and mirror symmetry (horizontal symmetry), change in contrast, sharpness and brightness). Moreover, the whole dataset was transformed into Gaussian noise and PCA jittering as well. Furthermore, the selection of dataset was explained by the help of plots to prove the significance

Plants **2019**, *8*, 468

of extending the dataset. A new CNN model named LeafNet was introduced in [88] to classify the tea leaf diseases and achieved higher accuracy than Support Vector Machine (SVM) and Multi-Layer Perceptron (MLP). In [89], two DL models named modified MobileNet and reduced MobileNet were introduced, and their accuracy was near to the VGG model; the reduced MobileNet actually got 98.34% classification accuracy and had a fewer number of parameters as compared to VGG which saves time in training the model. A state-of-the-art DL model was proposed in [90] named PlantdiseaseNet which was remarkably suitable for the complex environment of an agricultural field. In [85], five types of apple plant diseases were classified and detected by the state-of-the-art CNN model named VGG-inception architecture. It outclassed the performance of many DL architectures like AlexNet, GoogLeNet, several versions of ResNet, and VGG. It also presented inter object/class detection and activation visualization; it was also mentioned for its clear vision of diseases in the plants.

A bar chart presented in Figure 12 indicates, from the most to the least frequently used, DL models for plant disease detection and classification. It can be clearly seen that the AlexNet model has been used in most of the researches. GoogLeNet, VGG-16, and ResNet-50 are the next most commonly used DL models. Similarly, there are some improved/cascaded versions (Improved Cifar-10, VGG-Inception, Cascaded AlexNet with GoogLeNet, reduced/modified MobileNet, modified LeNet, and modified GoogLeNet), which have been used for plant disease identification.

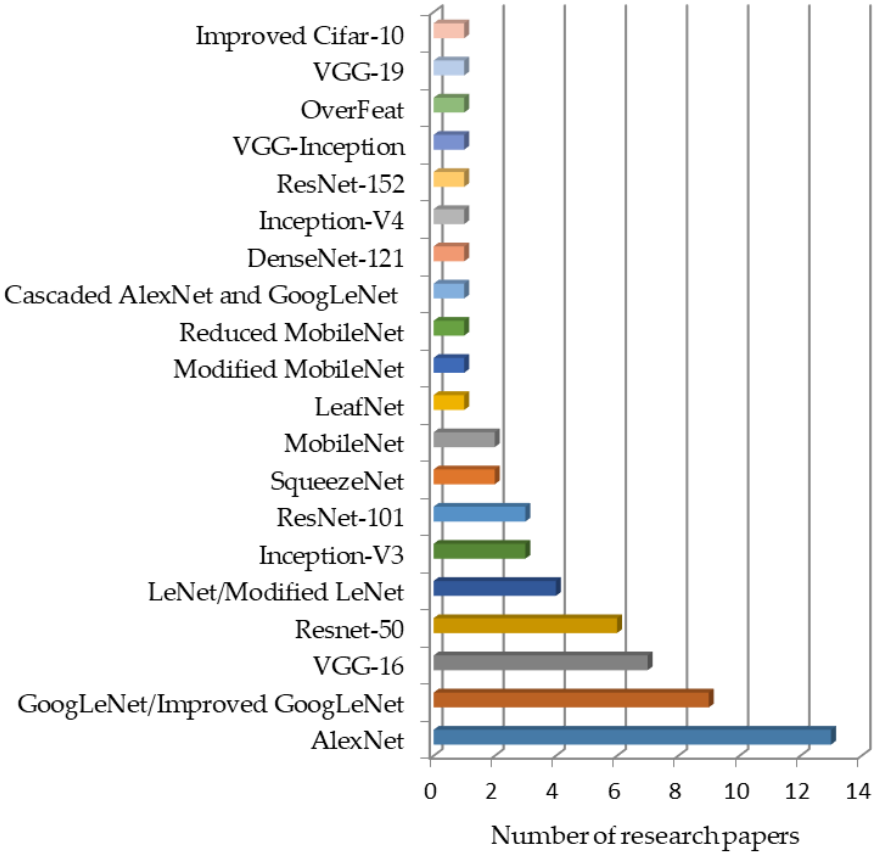


Figure 12. Deep learning models used in the particular number of research papers.

Summing up Section 2, all the DL approaches along with the selected plant species and performance metrics are shown in Table 3.

3. Hyper-Spectral Imaging with DL Models

For early detection of plant diseases, several imaging techniques like multispectral imaging [91], thermal imaging, fluorescence and hyperspectral imaging are used [92]. Among them, hyperspectral

imaging (HSI) is the focus of recent research. For example, [93] used hyperspectral imaging (HSI) to detect tomato plant diseases by identifying the region of interest, and a feature ranking-KNN (FR-KNN) model produced a satisfactory result for the detection of diseased and healthy plants. In the recent approach, HSI was used for the detection of an apple disease. Moreover, the redundancy issue was resolved by an unsupervised feature selection procedure known as Orthogonal Subspace Projection [94]. In [95], leaf diseases on peanuts were detected by HSI by identifying sensitive bands and hyperspectral vegetation index. The tomato disease detection was done by SVM classifiers based on HSI, and their performance was evaluated by F1-score, accuracy, specificity, and sensitivity [96].

Recently, HSI has been used with machine learning (ML) for the detection of plant diseases. For example, [97] described ML techniques for hyperspectral imaging for many agricultural applications. Moreover, ML with HSI have been used for three ML models, implemented by using hyperspectral measurement technique for the detection of leaf rust disease [98]. For wheat disease detection, [99] used Random Forest (RF) classifier with multispectral imaging technique and achieved accuracy of 89.3%. Plants' diseases were also detected by SVM based on hyperspectral data and achieved accuracy of more than 86% [100]. There are some other ML approaches based on HSI [101], but this review is focused on DL approaches based on HSI, presented below.

The DL has been used to classify the hyperspectral images for many applications. For medical purposes, this technology is very useful as it is used for the classification of head/neck cancer in [102]. In [103], a DL approach based on HSI was proposed through contextual information as it provides spectral and spatial features. A new 3D-CNN architecture allowed for a fast, accurate, and efficient approach to classify the hyperspectral images in [104]. This architecture not only used the spectral information (as used in previous CNN techniques [105]) but also ensured that the spatial information was also taken into account. In [106], the feature extraction procedure was used with CNN for hyperspectral image classification and used dropout and L2 regularization methods in order to prevent overfitting. Just like CNN models used for hyperspectral imaging classification, RNN models are also used with HSI as described in [107,108]. In the domain of plant disease detection, some researches utilized Hyperspectral Imaging (HSI) along with DL models to observe clearer vision for symptoms of plant diseases. A hybrid method to classify the hyperspectral images was proposed in [109] consisting of DCNN, LR, and PCA and got better results compared to the previous methods for classification tasks. In [110], a detailed review of DL with HSI technique was provided. In order to avoid the overfitting and improve accuracy, a detailed comparison provided between several DL models like 1D/2D-CNN (2D-CNN better result), LSTM/GRU (both faced overfitting), 2D-CNN-LSTM/GRU (still overfitting) was observed. Therefore, a new hybrid approach from Convolutional and Bidirectional Gated Recurrent Network named 2D-CNN-BidLSTM/GRU was proposed for the hyperspectral images, which resolved the problem of overfitting and achieved 0.75 F1-score and 0.73 accuracy for wheat diseases detection [111]. According to [112], a hyperspectral proximal-sensing procedure based on the newest DL technique named Generative Adversarial Nets (GAN) was proposed in order to detect tomato plant disease before its clear symptoms appeared (as shown in Figure 13). In [84], a 3D-CNN approach was proposed for hyperspectral images to identify the Charcoal rot disease in soybeans and the CNN model was evaluated by accuracy (95.76%) and F1-score (0.87). The saliency map visualization was used, and the most delicate wavelength resulted as 733 nm, which approximately lies in the region of the wavelength of NIR. For the detection of potato virus, [113] described it by DL on the hyperspectral images and achieved acceptable values of precision (0.78) and recall (0.88). In [114], a DL model named multiple Inception-Resnet model was developed by using both spatial and spectral data on hyperspectral UAV images to detect the yellow rust in wheat (as shown in Figure 14). This model achieved an 85% accuracy, which is quite a lot higher than the RF-classifier (77%).

Table 3. Comparison of several DL approaches in terms of various performance metrics.

DL Architectures/Algorithms	Datasets	Selected Plant/s	Performance Metrics (and Their Results)	Refs
CNN	PlantVillage	Maize	CA (92.85%)	[56]
AlexNet, GoogLeNet, ResNet	PlantVillage	Tomato	CA by ResNet which gave the best value (97.28%)	[57]
LeNet	PlantVillage	Banana	CA (98.61%), F1 (98.64%)	[32]
AlexNet, ALexNetOWTBn, GoogLeNet, Overfeat, VGG	PlantVillage and in-field images	Apple, blueberry, banana, cabbage, cassava, cantaloupe, celery, cherry, cucumber, corn, eggplant, gourd, grape, orange, onion	Success rate of VGG (99.53%) which is the best among all	[58]
AlexNet, VGG16, VGG 19, SqueezeNet, GoogLeNet, Inceptionv3, InceptionResNetv2, ResNet50, Resnet101 Inceptionv3	Real field dataset	Apricot, Walnut, Peach, Cherry	F1(97.14), Accuracy (97.86 ± 1.56) of ResNet	[35]
CNN	Experimental field dataset	Cassava	CA (93%)	[59]
Super-Resolution Convolutional Neural Network (SCRNN)	Images taken from the research center	Cucumber	CA (82.3%)	[60]
CaffeNet	PlantVillage	Tomato	Accuracy (~90%)	[61]
AlexNet and GoogLeNet	Downloaded from the internet	Pear, cherry, peach, apple, grapevine	Precision (96.3%)	[27]
AlexNet, GoogLeNet, VGG- 16, ResNet-50,101, ResNetXt-101, Faster RCNN, SSD, R-FCN, ZFNet	PlantVillage	Apple, blueberry, bell pepper, cherry, corn, peach, grape, raspberry, potato, squash, soybean, strawberry, tomato	CA (99.35%) of GoogLeNet	[25]
CNN	Image taken in real fields	Tomato	Precision (85.98%) of ResNet-50 with Region based Fully Convolutional Network(R-FCN)	[68]
DCNN	Bisque platform of Cy Verse	Maize	Accuracy (96.7%)	[70]
AlexNet, GoogLeNet	Images were taken in real field	Rice	Accuracy (95.48%)	[78]
VGG-FCN-VD16 and VGG-FCN-S	PlantVillage	Tomato	Accuracy (0.9918 ± 0.169) of GoogLeNet	[72]
VGG-A, CNN	Wheat Disease Database 2017	Wheat	Accuracy (97.95%) of VGG-FCN-VD16	[73]
AlexNet	Images were taken in real field	Radish	Accuracy (93.3%)	[74]
AlexNet and SqueezeNet v1.1	Images were taken in real field	Soybean	CA (94.13%)	[77]
DCNN, Random forest, Support Vector Machine and AlexNet	PlantVillage	Tomato	CA (95.65%) of AlexNet	[62]
	PlantVillage dataset, Forestry Image dataset and agricultural field in China	Cucumber	CA (93.4%) of DCNN	[66]

Table 3. *Cont.*

DL Architectures/Algorithms	Datasets	Selected Plant/s	Performance Metrics (and Their Results)	Refs
Teacher/student architecture	PlantVillage	Apple, bell pepper, blueberry, cherry, corn, orange, grape, potato, raspberry, peach, soybean, strawberry, tomato, squash	Training accuracy and loss (~99%,~0–0.5%), validation accuracy and loss (~95%, ~10%)	[67]
Improved GoogLeNet, Cifar-10	PlantVillage and various websites	Maize	Top-1 accuracy (98.9%) of improved GoogLeNet	[86]
MobileNet, Modified MobileNet, Reduced MobileNet	PlantVillage dataset	24 types of plant	CA (98.34%) of reduced MobileNet	[89]
VGG-16, ResNet-50,101,152, Inception-V4 and DenseNets-121	PlantVillage	Apple, bell pepper, blueberry, cherry, corn, orange, grape, potato, raspberry, peach, soybean, strawberry, tomato, squash	Testing accuracy (99.75%) of DenseNets	[63]
User defined CNN, SVM, AlexNet, GoogLeNet, ResNet-20 and VGG-16	Images were taken in real field	Apple	CA (97.62%) of proposed CNN	[87]
AlexNet and VGG-16	PlantVillage	Tomato	CA (AlexNet)	[64]
LeafNet, SVM, MLP	Images were taken in real field	Tea leaf	CA (90.16%) of LeafNet	[88]
2D-CNN-BidGRU	Real wheat field	wheat	F1 (0.75) and accuracy (0.743)	[111]
OR-AC-GAN	Real environment	Tomato	Accuracy (96.25%)	[112]
3D CNN	Real environment	Soybean	CA (95.73%), F1-score (0.87)	[84]
DCNN	Real environment	Wheat	Accuracy (85%)	[114]
ResNet-50	Real environment	Wheat	Balanced Accuracy (87%)	[79]
GPDCNN	Real environment	Cucumber	CA (94.65%)	[81]
VGG-16, AlexNet	PlantVillage, CASC-IFW	Apple, banana	CA (98.6%)	[82]
LeNet	Real environment	Grapes	CA (95.8%)	[83]
PlantDiseaseNet	Real environment	Apple, bell-pepper, cherry, grapes, onion, peach, potato, plum, strawberry, sugar-beets, tomato, wheat	CA (93.67%)	[90]
LeNet	PlantVillage	Soybean	CA (99.32%)	[71]
VGG-Inception	Real environment	Apple	Mean average accuracy (78.8%)	[85]
Resnet-50, Inception-V2, MobileNet-V1	Real environment	Banana	Mean average accuracy (99%) of ResNet-50	[69]
Modified LeNet	PlantVillage	Olives	True positive rate (98.6 ± 1.47%)	[65]

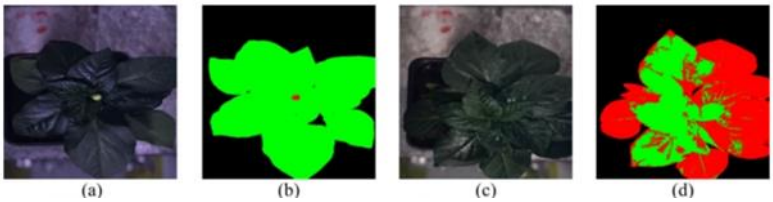


Figure 13. Sample images of OR-AC-GAN (a hyperspectral imaging model) [112].

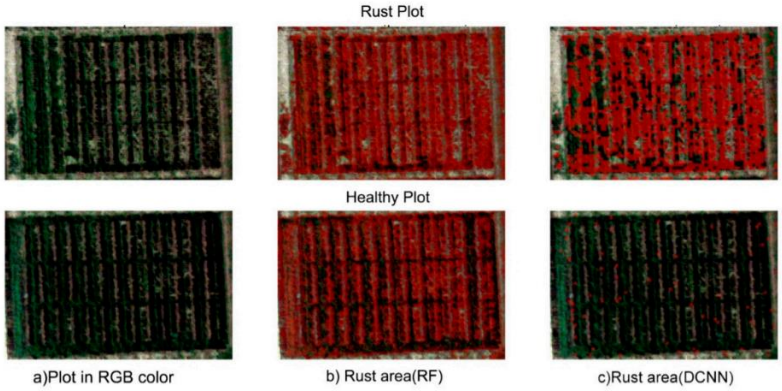


Figure 14. Hyperspectral images by UAV: (**a**) RGB color plots, (**b**) Random-Forest classifier, and (**c**) proposed multiple Inception-ResNet model [114].

From this section, we can conclude that, although there are some DL models/architectures developed for hyperspectral image classification in the application of plant disease detection, this is still a fertile area of research and should lead to improvements for better detection of plants' diseases [115] in different situations, like various conditions of illumination, considering real background, etc.

In Figure 13, the resultant images are taken from the proposed method described in [112]. The green-colored portion indicates the healthy part of the plant; the red portion denotes the infected portion. Note that (**a**) and (**b**) are the healthy plant images as there is no red color indication, whereas (**c**) has infected disease which can be seen in its corresponding figure (**d**).

A comparison of proposed DCNN with RF classifier and RGB colored hyperspectral images are shown in Figure 14. The red color label indicates the portion infected by rust. It should be observed that the rust plots were identified in an almost similar manner (see (b) and (c) of first row), but in the healthy plot, there was a large portion covered by the red label in (b) as compared to (c), which shows a wrong classification by RF model [114].

4. Conclusions and Future Directions

This review explained DL approaches for the detection of plant diseases. Moreover, many visualization techniques/mappings were summarized to recognize the symptoms of diseases. Although much significant progress was observed during the last three to four years, there are still some research gaps which are described below:

- In most of the researches (as described in the previous sections), the PlantVillage dataset was used to evaluate the accuracy and performance of the respective DL models/architectures. Although this dataset has a lot of images of several plant species with their diseases, it has a simple/plain background. However, for a practical scenario, the real environment should be considered.
- Hyperspectral/multispectral imaging is an emerging technology and has been used in many areas of research (as described in Section 3). Therefore, it should be used with the efficient DL architectures to detect the plants' diseases even before their symptoms are clearly apparent.
- A more efficient way of visualizing the spots of disease in plants should be introduced as it will save costs by avoiding the unnecessary application of fungicide/pesticide/herbicide.

Plants **2019**, *8*, 468

- The severity of plant diseases changes with the passage of time, therefore, DL models should be improved/modified to enable them to detect and classify diseases during their complete cycle of occurrence.
- DL model/architecture should be efficient for many illumination conditions, so the datasets should not only indicate the real environment but also contain images taken in different field scenarios.
- A comprehensive study is required to understand the factors affecting the detection of plant diseases, like the classes and size of datasets, learning rate, illumination, and the like.

Author Contributions: Conceptualization, M.H.S. and K.M.A.; methodology, M.H.S. and K.M.A.; writing—original draft preparation, M.H.S. and K.M.A.; writing—review and editing, M.H.S., J.P., and K.M.A.; visualization, M.H.S., J.P., and K.M.A.; supervision, J.P., and K.M.A.; project administration, J.P., and K.M.A.

Funding: This research was funded by the Ministry of Business, Innovation and Employment (MBIE), New Zealand, Science for Technological Innovation (SfTI) National Science Challenge.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The abbreviations used in this manuscript are given as under:

ML	Machine Learning
DL	Deep Learning
CNN	Convolutional Neural network
DCNN	Deep Convolutional Neural Network
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
RF	Random Forest
CA	Classification Accuracy
LSTM	Long Short-Term Memory
IoU	Intersection of Union
NiN	Network in Network
RCN	Region based Convolutional Neural Network
FCN	Fully Convolutional Neural Network
YOLO	You Only Look Once
SSD	Single Shot Detector
PSPNet	Pyramid Scene Parsing Network
IRRCNN	Inception Recurrent Residual Convolutional Neural Network
IRCNN	Inception Recurrent Convolutional Neural Network
DCRN	Densely Connected Recurrent Convolutional Network
INAR-SSD	Single Shot Detector with Inception module and Rainbow concatenation
R2U-Net	Recurrent Residual Convolutional Neural Network based on U-Net model
SVM	Support Vector Machines
ELM	Extreme Learning Machine
KNN	K-Nearest Neighbor
SRCNN	Super-Resolution Convolutional Neural Network
R-FCN	Region-based Fully Convolutional Networks
ROC	Receiver Operating Characteristic
PCA	Principal Component Analysis
MLP	Multi-Layer Perceptron
LRP	Layer-wise Relevance Propagation
HSI	Hyperspectral Imaging
FRKNN	Feature Ranking K-Nearest Neighbor

RNN	Recurrent Neural Network
ToF	Time-of-Flight
LR	Logistic Regression
GRU	Gated Recurrent Unit
AN	Generative Adversarial Nets
GPDCNN	Global Pooling Dilated Convolutional Neural Network
2D-CNN-BidGRU	2D-Convolutional-Bidirectional Gated Recurrent Unit Neural Network
OR-AC-GAN	Outlier Removal-Auxiliary Classifier-Generative Adversarial Nets

References

1. McCulloch, W.S.; Pitts, W. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* **1943**, *5*, 115–133. [CrossRef]
2. Ackley, D.H.; Hinton, G.E.; Sejnowski, T.J. A learning algorithm for Boltzmann machines. *Cogn. Sci.* **1985**, *9*, 147–169. [CrossRef]
3. Kelley, H.J. Gradient theory of optimal flight paths. *Ars J.* **1960**, *30*, 947–954. [CrossRef]
4. Dreyfus, S. The numerical solution of variational problems. *J. Math. Anal. Appl.* **1962**, *5*, 30–45. [CrossRef]
5. Fukushima, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cybern.* **1980**, *36*, 193–202. [CrossRef]
6. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
7. Hinton, G.E.; Osindero, S.; Teh, Y.-W. A fast learning algorithm for deep belief nets. *Neural Comput.* **2006**, *18*, 1527–1554. [CrossRef]
8. Hinton, G.E.; Salakhutdinov, R.R. Reducing the dimensionality of data with neural networks. *Science* **2006**, *313*, 504–507. [CrossRef]
9. Fayjie, A.R.; Hossain, S.; Oualid, D.; Lee, D.-J. Driverless Car: Autonomous Driving Using Deep Reinforcement Learning in Urban Environment. In Proceedings of the 2018 15th International Conference on Ubiquitous Robots (UR), Hawaii Convention Center, Honolulu, HI, USA, 26–30 June 2018; pp. 896–901.
10. Hossain, S.; Lee, D.-J. Autonomous-Driving Vehicle Learning Environments using Unity Real-time Engine and End-to-End CNN Approach. *J. Korea Robot. Soc.* **2019**, *14*, 122–130. [CrossRef]
11. Kocić, J.; Jovičić, N.; Drndarević, V. An End-to-End Deep Neural Network for Autonomous Driving Designed for Embedded Automotive Platforms. *Sensors* **2019**, *19*, 2064. [CrossRef]
12. Esteva, A.; Robicquet, A.; Ramsundar, B.; Kuleshov, V.; DePristo, M.; Chou, K.; Cui, C.; Corrado, G.; Thrun, S.; Dean, J. A guide to deep learning in healthcare. *Nat. Med.* **2019**, *25*, 24. [CrossRef] [PubMed]
13. Miotto, R.; Wang, F.; Wang, S.; Jiang, X.; Dudley, J.T. Deep learning for healthcare: Review, opportunities and challenges. *Brief. Bioinform.* **2017**, *19*, 1236–1246. [CrossRef] [PubMed]
14. Ravì, D.; Wong, C.; Deligianni, F.; Berthelot, M.; Andreu-Perez, J.; Lo, B.; Yang, G.-Z. Deep learning for health informatics. *IEEE J. Biomed. Health Inform.* **2016**, *21*, 4–21. [CrossRef] [PubMed]
15. Goodfellow, I.J.; Bulatov, Y.; Ibarz, J.; Arnoud, S.; Shet, V. Multi-digit number recognition from street view imagery using deep convolutional neural networks. *arXiv* **2013**, arXiv:1312.6082.
16. Jaderberg, M.; Simonyan, K.; Vedaldi, A.; Zisserman, A. Deep structured output learning for unconstrained text recognition. *arXiv* **2014**, arXiv:1412.5903.
17. Yousfi, S.; Berrani, S.-A.; Garcia, C. Deep learning and recurrent connectionist-based approaches for Arabic text recognition in videos. In Proceedings of the 2015 13th International Conference on Document Analysis and Recognition (ICDAR), Tunis, Tunisia, 23–26 August 2015; pp. 1026–1030.
18. DeVries, P.M.; Viégas, F.; Wattenberg, M.; Meade, B.J. Deep learning of aftershock patterns following large earthquakes. *Nature* **2018**, *560*, 632. [CrossRef]
19. Mousavi, S.M.; Zhu, W.; Sheng, Y.; Beroza, G.C. CRED: A deep residual network of convolutional and recurrent units for earthquake signal detection. *Sci. Rep.* **2019**, *9*, 10267. [CrossRef]
20. Perol, T.; Gharbi, M.; Denolle, M. Convolutional neural network for earthquake detection and location. *Sci. Adv.* **2018**, *4*, e1700578. [CrossRef]

21. Siau, K.; Yang, Y. Impact of artificial intelligence, robotics, and machine learning on sales and marketing. In Proceedings of the Twelve Annual Midwest Association for Information Systems Conference (MWAIS 2017), Springfield, IL, USA, 18–19 May 2017; pp. 18–19.
22. Heaton, J.; Polson, N.; Witte, J.H. Deep learning for finance: Deep portfolios. *Appl. Stoch. Models Bus. Ind.* **2017**, *33*, 3–12. [CrossRef]
23. Heaton, J.; Polson, N.G.; Witte, J.H. Deep learning in finance. *arXiv* **2016**, arXiv:1602.06561.
24. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
25. Mohanty, S.P.; Hughes, D.P.; Salathé, M. Using deep learning for image-based plant disease detection. *Front. Plant Sci.* **2016**, *7*, 1419. [CrossRef]
26. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
27. Sladojevic, S.; Arsenovic, M.; Anderla, A.; Culibrk, D.; Stefanovic, D. Deep neural networks based recognition of plant diseases by leaf image classification. *Comput. Intell. Neurosci.* **2016**, *2016*. [CrossRef]
28. Wan, J.; Wang, D.; Hoi, S.C.H.; Wu, P.; Zhu, J.; Zhang, Y.; Li, J. Deep learning for content-based image retrieval: A comprehensive study. In Proceedings of the 22nd ACM International Conference on Multimedia, Orlando, FL, USA, 3–7 November 2014; pp. 157–166.
29. Wu, R.; Yan, S.; Shan, Y.; Dang, Q.; Sun, G. Deep image: Scaling up image recognition. *arXiv* **2015**, arXiv:1501.02876.
30. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–8 December 2012; pp. 1097–1105.
31. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
32. Amara, J.; Bouaziz, B.; Algergawy, A. A Deep Learning-based Approach for Banana Leaf Diseases Classification. In Proceedings of the BTW (Workshops), Stuttgart, Germany, 6–10 March 2017; pp. 79–88.
33. Rebetez, J.; Satizábal, H.F.; Mota, M.; Noll, D.; Büchi, L.; Wendling, M.; Cannelle, B.; Pérez-Uribe, A.; Burgos, S. Augmenting a convolutional neural network with local histograms—A case study in crop classification from high-resolution UAV imagery. In Proceedings of the ESANN, Bruges, Belgium, 27–29 April 2016.
34. Rußwurm, M.; Körner, M. Multi-temporal land cover classification with long short-term memory neural networks. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *42*, 551. [CrossRef]
35. TÜRKOĞLU, M.; Hanbay, D. Plant disease and pest detection using deep learning-based features. *Turk. J. Electr. Eng. Comput. Sci.* **2019**, *27*, 1636–1651. [CrossRef]
36. Mortensen, A.K.; Dyrmann, M.; Karstoft, H.; Jørgensen, R.N.; Gislum, R. Semantic segmentation of mixed crops using deep convolutional neural network. In Proceedings of the CIGR-AgEng Conference, Aarhus, Denmark, 26–29 June 2016; Abstracts and Full Papers. pp. 1–6.
37. Dyrmann, M.; Karstoft, H.; Midtiby, H.S. Plant species classification using deep convolutional neural network. *Biosyst. Eng.* **2016**, *151*, 72–80. [CrossRef]
38. McCool, C.; Perez, T.; Upcroft, B. Mixtures of lightweight deep convolutional neural networks: Applied to agricultural robotics. *IEEE Robot. Autom. Lett.* **2017**, *2*, 1344–1351. [CrossRef]
39. Santoni, M.M.; Sensuse, D.I.; Arymurthy, A.M.; Fanany, M.I. Cattle race classification using gray level co-occurrence matrix convolutional neural networks. *Procedia Comput. Sci.* **2015**, *59*, 493–502. [CrossRef]
40. Sørensen, R.A.; Rasmussen, J.; Nielsen, J.; Jørgensen, R.N. Thistle detection using convolutional neural networks. In Proceedings of the 2017 EFITA WCCA CONGRESS, Montpellier, France, 2–6 July 2017; p. 161.
41. Xinshao, W.; Cheng, C. Weed seeds classification based on PCANet deep learning baseline. In Proceedings of the 2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA), Hong Kong, China, 16–19 December 2015; pp. 408–415.
42. Hall, D.; McCool, C.; Dayoub, F.; Sunderhauf, N.; Upcroft, B. Evaluation of features for leaf classification in challenging conditions. In Proceedings of the 2015 IEEE Winter Conference on Applications of Computer Vision, Waikoloa Beach, HI, USA, 6–8 January 2015; pp. 797–804.

43. Kamilaris, A.; Prenafeta-Boldú, F.X. Deep learning in agriculture: A survey. *Comput. Electron. Agric.* **2018**, *147*, 70–90. [CrossRef]
44. Itzhaky, Y.; Farjon, G.; Khoroshevsky, F.; Shpigler, A.; Bar-Hillel, A. Leaf counting: Multiple scale regression and detection using deep CNNs. In Proceedings of the BMVC, North East, UK, 3–6 September 2018; p. 328.
45. Ubbens, J.; Cieslak, M.; Prusinkiewicz, P.; Stavness, I. The use of plant models in deep learning: An application to leaf counting in rosette plants. *Plant Methods* **2018**, *14*, 6. [CrossRef]
46. Rahnemoonfar, M.; Sheppard, C. Deep count: Fruit counting based on deep simulated learning. *Sensors* **2017**, *17*, 905. [CrossRef]
47. Kussul, N.; Lavreniuk, M.; Skakun, S.; Shelestov, A. Deep learning classification of land cover and crop types using remote sensing data. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 778–782. [CrossRef]
48. Grinblat, G.L.; Uzal, L.C.; Larese, M.G.; Granitto, P.M. Deep learning for plant identification using vein morphological patterns. *Comput. Electron. Agric.* **2016**, *127*, 418–424. [CrossRef]
49. Lee, S.H.; Chan, C.S.; Wilkin, P.; Remagnino, P. Deep-plant: Plant identification with convolutional neural networks. In Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP), Québec City, QC, Canada, 27–30 September 2015; pp. 452–456.
50. Pound, M.P.; Atkinson, J.A.; Townsend, A.J.; Wilson, M.H.; Griffiths, M.; Jackson, A.S.; Bulat, A.; Tzimiropoulos, G.; Wells, D.M.; Murchie, E.H. Deep machine learning provides state-of-the-art performance in image-based plant phenotyping. *Gigascience* **2017**, *6*, gix083. [CrossRef] [PubMed]
51. Milioto, A.; Lottes, P.; Stachniss, C. Real-time blob-wise sugar beets vs weeds classification for monitoring fields using convolutional neural networks. *Isprs Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *4*, 41. [CrossRef]
52. Potena, C.; Nardi, D.; Pretto, A. Fast and accurate crop and weed identification with summarized train sets for precision agriculture. In Proceedings of the International Conference on Intelligent Autonomous Systems, Shanghai, China, 3–7 July 2016; pp. 105–121.
53. Sun, Y.; Liu, Y.; Wang, G.; Zhang, H. Deep learning for plant identification in natural environment. *Comput. Intell. Neurosci.* **2017**, *2017*, 7361042. [CrossRef] [PubMed]
54. Singh, A.K.; Ganapathysubramanian, B.; Sarkar, S.; Singh, A. Deep learning for plant stress phenotyping: Trends and future perspectives. *Trends Plant Sci.* **2018**, *23*, 883–898. [CrossRef] [PubMed]
55. Brahimi, M.; Arsenovic, M.; Laraba, S.; Sladojevic, S.; Boukhalfa, K.; Moussaoui, A. Deep learning for plant diseases: Detection and saliency map visualisation. In *Human and Machine Learning*; Springer: Berlin, Germany, 2018; pp. 93–117.
56. Sibiya, M.; Sumbwanyambe, M. A Computational Procedure for the Recognition and Classification of Maize Leaf Diseases Out of Healthy Leaves Using Convolutional Neural Networks. *AgriEngineering* **2019**, *1*, 119–131. [CrossRef]
57. Zhang, K.; Wu, Q.; Liu, A.; Meng, X. Can Deep Learning Identify Tomato Leaf Disease? *Adv. Multimed.* **2018**, *2018*, 10. [CrossRef]
58. Ferentinos, K.P. Deep learning models for plant disease detection and diagnosis. *Comput. Electron. Agric.* **2018**, *145*, 311–318. [CrossRef]
59. Ramcharan, A.; Baranowski, K.; McCloskey, P.; Ahmed, B.; Legg, J.; Hughes, D.P. Deep learning for image-based cassava disease detection. *Front. Plant Sci.* **2017**, *8*, 1852. [CrossRef]
60. Fujita, E.; Kawasaki, Y.; Uga, H.; Kagiwada, S.; Iyatomi, H. Basic investigation on a robust and practical plant diagnostic system. In Proceedings of the 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA), Anaheim, CA, USA, 18–20 December 2016; pp. 989–992.
61. Yamamoto, K.; Togami, T.; Yamaguchi, N. Super-resolution of plant disease images for the acceleration of image-based phenotyping and vigor diagnosis in agriculture. *Sensors* **2017**, *17*, 2557. [CrossRef]
62. Durmuş, H.; Güneş, E.O.; Kırcı, M. Disease detection on the leaves of the tomato plants by using deep learning. In Proceedings of the 2017 6th International Conference on Agro-Geoinformatics, Fairfax, VA, USA, 7–10 August 2017; pp. 1–5.
63. Too, E.C.; Yujian, L.; Njuki, S.; Yingchun, L. A comparative study of fine-tuning deep learning models for plant disease identification. *Comput. Electron. Agric.* **2019**, *161*, 272–279. [CrossRef]
64. Rangarajan, A.K.; Purushothaman, R.; Ramesh, A. Tomato crop disease classification using pre-trained deep learning algorithm. *Procedia Comput. Sci.* **2018**, *133*, 1040–1047. [CrossRef]

65. Cruz, A.C.; Luvisi, A.; De Bellis, L.; Ampatzidis, Y. Vision-based plant disease detection system using transfer and deep learning. In Proceedings of the 2017 ASABE Annual International Meeting, Spokane, WA, USA, 16–19 July 2017; p. 1.
66. Ma, J.; Du, K.; Zheng, F.; Zhang, L.; Gong, Z.; Sun, Z. A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network. *Comput. Electron. Agric.* **2018**, *154*, 18–24. [CrossRef]
67. Brahimi, M.; Mahmoudi, S.; Boukhalfa, K.; Moussaoui, A. Deep interpretable architecture for plant diseases classification. *arXiv* **2019**, arXiv:1905.13523.
68. Fuentes, A.; Yoon, S.; Kim, S.; Park, D. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors* **2017**, *17*, 2022. [CrossRef] [PubMed]
69. Selvaraj, M.G.; Vergara, A.; Ruiz, H.; Safari, N.; Elayabalan, S.; Ocimati, W.; Blomme, G. AI-powered banana diseases and pest detection. *Plant Methods* **2019**, *15*, 92. [CrossRef]
70. DeChant, C.; Wiesner-Hanks, T.; Chen, S.; Stewart, E.L.; Yosinski, J.; Gore, M.A.; Nelson, R.J.; Lipson, H. Automated identification of northern leaf blight-infected maize plants from field imagery using deep learning. *Phytopathology* **2017**, *107*, 1426–1432. [CrossRef]
71. Wallelign, S.; Polceanu, M.; Buche, C. Soybean Plant Disease Identification Using Convolutional Neural Network. In Proceedings of the Thirty-First International Flairs Conference, Melbourne, FL, USA, 21–23 May 2018.
72. Brahimi, M.; Boukhalfa, K.; Moussaoui, A. Deep learning for tomato diseases: Classification and symptoms visualization. *Appl. Artif. Intell.* **2017**, *31*, 299–315. [CrossRef]
73. Lu, J.; Hu, J.; Zhao, G.; Mei, F.; Zhang, C. An in-field automatic wheat disease diagnosis system. *Comput. Electron. Agric.* **2017**, *142*, 369–379. [CrossRef]
74. Ha, J.G.; Moon, H.; Kwak, J.T.; Hassan, S.I.; Dang, M.; Lee, O.N.; Park, H.Y. Deep convolutional neural network for classifying Fusarium wilt of radish from unmanned aerial vehicles. *J. Appl. Remote Sens.* **2017**, *11*, 042621. [CrossRef]
75. Lin, K.; Gong, L.; Huang, Y.; Liu, C.; Pan, J. Deep learning-based segmentation and quantification of cucumber Powdery Mildew using convolutional neural network. *Front. Plant Sci.* **2019**, *10*, 155. [CrossRef]
76. Barbedo, J.G.A. Plant disease identification from individual lesions and spots using deep learning. *Biosyst. Eng.* **2019**, *180*, 96–107. [CrossRef]
77. Ghosal, S.; Blystone, D.; Singh, A.K.; Ganapathysubramanian, B.; Singh, A.; Sarkar, S. An explainable deep machine vision framework for plant stress phenotyping. *Proc. Natl. Acad. Sci.* **2018**, *115*, 4613–4618. [CrossRef] [PubMed]
78. Lu, Y.; Yi, S.; Zeng, N.; Liu, Y.; Zhang, Y. Identification of rice diseases using deep convolutional neural networks. *Neurocomputing* **2017**, *267*, 378–384. [CrossRef]
79. Picon, A.; Alvarez-Gila, A.; Seitz, M.; Ortiz-Barredo, A.; Echazarra, J.; Johannes, A. Deep convolutional neural networks for mobile capture device-based crop disease classification in the wild. *Comput. Electron. Agric.* **2019**, *161*, 280–290. [CrossRef]
80. Johannes, A.; Picon, A.; Alvarez-Gila, A.; Echazarra, J.; Rodriguez-Vaamonde, S.; Navajas, A.D.; Ortiz-Barredo, A. Automatic plant disease diagnosis using mobile capture devices, applied on a wheat use case. *Comput. Electron. Agric.* **2017**, *138*, 200–209. [CrossRef]
81. Zhang, S.; Zhang, S.; Zhang, C.; Wang, X.; Shi, Y. Cucumber leaf disease identification with global pooling dilated convolutional neural network. *Comput. Electron. Agric.* **2019**, *162*, 422–430. [CrossRef]
82. Khan, M.A.; Akram, T.; Sharif, M.; Awais, M.; Javed, K.; Ali, H.; Saba, T. CCDF: Automatic system for segmentation and recognition of fruit crops diseases based on correlation coefficient and deep CNN features. *Comput. Electron. Agric.* **2018**, *155*, 220–236. [CrossRef]
83. Kerkech, M.; Hafiane, A.; Canals, R. Deep leaning approach with colorimetric spaces and vegetation indices for vine diseases detection in UAV images. *Comput. Electron. Agric.* **2018**, *155*, 237–243. [CrossRef]
84. Nagasubramanian, K.; Jones, S.; Singh, A.K.; Singh, A.; Ganapathysubramanian, B.; Sarkar, S. Explaining hyperspectral imaging based plant disease identification: 3D CNN and saliency maps. *arXiv* **2018**, arXiv:1804.08831.
85. Jiang, P.; Chen, Y.; Liu, B.; He, D.; Liang, C. Real-Time Detection of Apple Leaf Diseases Using Deep Learning Approach Based on Improved Convolutional Neural Networks. *IEEE Access* **2019**, *7*, 59069–59080. [CrossRef]

86. Zhang, X.; Qiao, Y.; Meng, F.; Fan, C.; Zhang, M. Identification of maize leaf diseases using improved deep convolutional neural networks. *IEEE Access* **2018**, *6*, 30370–30377. [CrossRef]
87. Liu, B.; Zhang, Y.; He, D.; Li, Y. Identification of apple leaf diseases based on deep convolutional neural networks. *Symmetry* **2017**, *10*, 11. [CrossRef]
88. Chen, J.; Liu, Q.; Gao, L. Visual Tea Leaf Disease Recognition Using a Convolutional Neural Network Model. *Symmetry* **2019**, *11*, 343. [CrossRef]
89. Kamal, K.; Yin, Z.; Wu, M.; Wu, Z. Depthwise separable convolution architectures for plant disease classification. *Comput. Electron. Agric.* **2019**, *165*, 104948.
90. Arsenovic, M.; Karanovic, M.; Sladojevic, S.; Anderla, A.; Stefanovic, D. Solving Current Limitations of Deep Learning Based Approaches for Plant Disease Detection. *Symmetry* **2019**, *11*, 939. [CrossRef]
91. Veys, C.; Chatziavgerinos, F.; AlSuwaidi, A.; Hibbert, J.; Hansen, M.; Bernotas, G.; Smith, M.; Yin, H.; Rolfe, S.; Grieve, B. Multispectral imaging for presymptomatic analysis of light leaf spot in oilseed rape. *Plant Methods* **2019**, *15*, 4. [CrossRef] [PubMed]
92. Mahlein, A.-K.; Alisaac, E.; Al Masri, A.; Behmann, J.; Dehne, H.-W.; Oerke, E.-C. Comparison and Combination of Thermal, Fluorescence, and Hyperspectral Imaging for Monitoring Fusarium Head Blight of Wheat on Spikelet Scale. *Sensors* **2019**, *19*, 2281. [CrossRef] [PubMed]
93. Xie, C.; Yang, C.; He, Y. Hyperspectral imaging for classification of healthy and gray mold diseased tomato leaves with different infection severities. *Comput. Electron. Agric.* **2017**, *135*, 154–162. [CrossRef]
94. Shuaibu, M.; Lee, W.S.; Schueller, J.; Gader, P.; Hong, Y.K.; Kim, S. Unsupervised hyperspectral band selection for apple Marssonina blotch detection. *Comput. Electron. Agric.* **2018**, *148*, 45–53. [CrossRef]
95. Chen, T.; Zhang, J.; Chen, Y.; Wan, S.; Zhang, L. Detection of peanut leaf spots disease using canopy hyperspectral reflectance. *Comput. Electron. Agric.* **2019**, *156*, 677–683. [CrossRef]
96. Moghadam, P.; Ward, D.; Goan, E.; Jayawardena, S.; Sikka, P.; Hernandez, E. Plant disease detection using hyperspectral imaging. In Proceedings of the 2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA), Sydney, Australia, 29 November–1 December 2017; pp. 1–8.
97. Hruška, J.; Adão, T.; Pádua, L.; Marques, P.; Cunha, A.; Peres, E.; Sousa, A.; Morais, R.; Sousa, J.J. Machine learning classification methods in hyperspectral data processing for agricultural applications. In Proceedings of the International Conference on Geoinformatics and Data Analysis, Prague, Czech Republic, 20–22 April 2018; pp. 137–141.
98. Ashourloo, D.; Aghighi, H.; Matkan, A.A.; Mobasheri, M.R.; Rad, A.M. An investigation into machine learning regression techniques for the leaf rust disease detection using hyperspectral measurement. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2016**, *9*, 4344–4351. [CrossRef]
99. Su, J.; Liu, C.; Coombes, M.; Hu, X.; Wang, C.; Xu, X.; Li, Q.; Guo, L.; Chen, W.-H. Wheat yellow rust monitoring by learning from multispectral UAV aerial imagery. *Comput. Electron. Agric.* **2018**, *155*, 157–166. [CrossRef]
100. Rumpf, T.; Mahlein, A.-K.; Steiner, U.; Oerke, E.-C.; Dehne, H.-W.; Plümer, L. Early detection and classification of plant diseases with support vector machines based on hyperspectral reflectance. *Comput. Electron. Agric.* **2010**, *74*, 91–99. [CrossRef]
101. Zhu, H.; Chu, B.; Zhang, C.; Liu, F.; Jiang, L.; He, Y. Hyperspectral imaging for presymptomatic detection of tobacco disease with successive projections algorithm and machine-learning classifiers. *Sci. Rep.* **2017**, *7*, 4125. [CrossRef] [PubMed]
102. Halicek, M.; Lu, G.; Little, J.V.; Wang, X.; Patel, M.; Griffith, C.C.; El-Deiry, M.W.; Chen, A.Y.; Fei, B. Deep convolutional neural networks for classifying head and neck cancer using hyperspectral imaging. *J. Biomed. Opt.* **2017**, *22*, 060503. [CrossRef] [PubMed]
103. Ma, X.; Geng, J.; Wang, H. Hyperspectral image classification via contextual deep learning. *Eurasip J. Image Video Process.* **2015**, *2015*, 20. [CrossRef]
104. Paoletti, M.; Haut, J.; Plaza, J.; Plaza, A. A new deep convolutional neural network for fast hyperspectral image classification. *Isprs J. Photogramm. Remote Sens.* **2018**, *145*, 120–147. [CrossRef]
105. Hu, W.; Huang, Y.; Wei, L.; Zhang, F.; Li, H. Deep convolutional neural networks for hyperspectral image classification. *J. Sens.* **2015**, *2015*. [CrossRef]
106. Chen, Y.; Jiang, H.; Li, C.; Jia, X.; Ghamisi, P. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 6232–6251. [CrossRef]

107. Mou, L.; Ghamisi, P.; Zhu, X.X. Deep recurrent neural networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 3639–3655. [CrossRef]
108. Wu, H.; Prasad, S. Convolutional recurrent neural networks forhyperspectral data classification. *Remote Sens.* **2017**, *9*, 298. [CrossRef]
109. Yue, J.; Zhao, W.; Mao, S.; Liu, H. Spectral–spatial classification of hyperspectral images using deep convolutional neural networks. *Remote Sens. Lett.* **2015**, *6*, 468–477. [CrossRef]
110. Signoroni, A.; Savardi, M.; Baronio, A.; Benini, S. Deep Learning Meets Hyperspectral Image Analysis: A Multidisciplinary Review. *J. Imaging* **2019**, *5*, 52. [CrossRef]
111. Jin, X.; Jie, L.; Wang, S.; Qi, H.; Li, S. Classifying wheat hyperspectral pixels of healthy heads and Fusarium head blight disease using a deep neural network in the wild field. *Remote Sens.* **2018**, *10*, 395. [CrossRef]
112. Wang, D.; Vinson, R.; Holmes, M.; Seibel, G.; Bechar, A.; Nof, S.; Tao, Y. Early Detection of Tomato Spotted Wilt Virus by Hyperspectral Imaging and Outlier Removal Auxiliary Classifier Generative Adversarial Nets (OR-AC-GAN). *Sci. Rep.* **2019**, *9*, 4377. [CrossRef] [PubMed]
113. Polder, G.; Blok, P.M.; de Villiers, H.A.C.; van der Wolf, J.M.; Kamp, J. Potato Virus Y Detection in Seed Potatoes Using Deep Learning on Hyperspectral Images. *Front. Plant Sci.* **2019**, *10*. [CrossRef] [PubMed]
114. Zhang, X.; Han, L.; Dong, Y.; Shi, Y.; Huang, W.; Han, L.; González-Moreno, P.; Ma, H.; Ye, H.; Sobeih, T. A Deep Learning-Based Approach for Automated Yellow Rust Disease Detection from High-Resolution Hyperspectral UAV Images. *Remote Sens.* **2019**, *11*, 1554. [CrossRef]
115. Golhani, K.; Balasundram, S.K.; Vadamalai, G.; Pradhan, B. A review of neural networks in plant disease detection using hyperspectral data. *Inf. Process. Agric.* **2018**, *5*, 354–371. [CrossRef]

Chapter 3. Plant Disease Classification: A Comparative Evaluation of Convolutional Neural Networks and Deep Learning Optimizers

This chapter contains content from the following article.

M. H. Saleem, J. Potgieter, and K. M. Arif, "Plant disease classification: A comparative evaluation of convolutional neural networks and deep learning optimizers," *Plants*, vol. 9, no. 10, p. 1319, 2020. https://doi.org/10.3390/plants9101319

plants

MDPI

Article

Plant Disease Classification: A Comparative Evaluation of Convolutional Neural Networks and Deep Learning Optimizers

Muhammad Hammad Saleem [1], Johan Potgieter [2] and Khalid Mahmood Arif [1,*]

[1] Department of Mechanical and Electrical Engineering, School of Food and Advanced Technology, Massey University, Auckland 0632, New Zealand; H.Saleem@massey.ac.nz
[2] Massey Agritech Partnership Research Centre, School of Food and Advanced Technology, Massey University, Palmerston North 4442, New Zealand; J.Potgieter@massey.ac.nz
* Correspondence: K.Arif@massey.ac.nz

Received: 9 September 2020; Accepted: 4 October 2020; Published: 6 October 2020

Abstract: Recently, plant disease classification has been done by various state-of-the-art deep learning (DL) architectures on the publicly available/author generated datasets. This research proposed the deep learning-based comparative evaluation for the classification of plant disease in two steps. Firstly, the best convolutional neural network (CNN) was obtained by conducting a comparative analysis among well-known CNN architectures along with modified and cascaded/hybrid versions of some of the DL models proposed in the recent researches. Secondly, the performance of the best-obtained model was attempted to improve by training through various deep learning optimizers. The comparison between various CNNs was based on performance metrics such as validation accuracy/loss, F1-score, and the required number of epochs. All the selected DL architectures were trained in the PlantVillage dataset which contains 26 different diseases belonging to 14 respective plant species. Keras with TensorFlow backend was used to train deep learning architectures. It is concluded that the Xception architecture trained with the Adam optimizer attained the highest validation accuracy and F1-score of 99.81% and 0.9978 respectively which is comparatively better than the previous approaches and it proves the novelty of the work. Therefore, the method proposed in this research can be applied to other agricultural applications for transparent detection and classification purposes.

Keywords: plant disease classification; convolutional neural network; deep learning; validation accuracy; F1-score

1. Introduction

In order to match the food demand, agricultural problems should be addressed by advanced techniques. In this regard, the agricultural industries are focusing on artificial intelligence methods. Several traditional machine learning (ML) algorithms have been used to perform various agricultural operations. On top of that, deep learning (DL) produced significant developments in the agricultural field of research. This is due to the automatic feature extraction capability of the deep learning algorithms. Among several agricultural problems, the successful classification of plant diseases is vital to improve the quality/quantity of agricultural products and reduce an undesirable application of chemical sprayers such as fungicide/herbicide. Therefore, it is an emerging research topic to advance agricultural automation. This agricultural task has a complexity due to the resemblance in the occurrence of the plant containing diseases. In this regard, several studies have been conducted to improve the classification of plant disease.

Many conventional machine learning (ML) models have been applied for plant disease classification [1,2]. Similarly, advanced imaging techniques including hyperspectral [3–7] and multispectral imaging [8] have also been used for plant/leaf disease identification. However, after the evolution of deep learning (DL), many state-of-the-art architectures, including AlexNet [9–14], Visual Geometry Group (VGG) [10,11,13,15,16], DenseNet [16], Inception-v4 [16] and ResNet [11,13,14,16], got promising results for the classification of plant disease. In this regard, several studies proved the significance of deep learning-based methods as compared to the traditional ML techniques. For example, a well-known DL model named GoogLeNet outperformed the ML algorithms including Support Vector Machine (SVM) and Random Forest (RF) models for the classification of disease in tomato leaves [9]. Another research showed the effectiveness of Convolutional Neural Networks (CNN) in comparison with the other state-of-the-art techniques such as Radial Basis Function Neural Network (RBFNN), Particle Swarm Optimization (PSO), and SVM for the classification of defects in mango leaves [17]. An article proposed a CNN model to identify diseases in apple leaves, which provided higher accuracy than SVM, Back-Propagation Neural Network model (BPNN), AlexNet, GoogLeNet, ResNet, and VGG models [18]. In [19], a CNN model was proposed to classify the disease in the leaves of PlantVillage dataset; its performance was better than the ML techniques such as SVM, Decision Tree (DT), Logistics Regression (LR), and K-Nearest Neighbor (KNN) models. This model also performed better than the well-known DL architectures including AlexNet, ResNet, VGG-16, and Inception-v3. Therefore, this article focuses on the DL-based models for the classification of plant disease.

Different approaches have been adopted to enhance the results of plant disease classification including modified versions of well-known DL models, various training techniques, data augmentation techniques, cascaded versions of two successful DL architectures, etc. [6]. For example, the famous GoogLeNet model was improved to achieve better testing accuracy for the identification of maize leaf disease in a small period due to its lesser number of parameters [20]. Similarly, inspired by the AlexNet model, a modified CNN architecture was proposed that had a lesser number of filters in convolutional layers and number of nodes, which apparently reduced overall parameters as compared to the original model and successfully identified the disease in tea leaves [21]. By using an extended version of the PlantVillage dataset, two modified versions of MobileNet models were proposed and their performance was compared with the original model (MobileNet), AlexNet, and VGG models [22]. Another research proposed a cascaded version of DL architecture to classify disease in apple leaves and it achieved better results as compared to AlexNet, GoogLeNet, VGG-16, Inception-v3, and various versions of ResNet models [23]. Moreover, several visualization techniques were also utilized along with DL models to highlight the disease spots in several plant species [9,24–26]. Few studies have been conducted to further advance the research of plant disease classification by using various training techniques. In [12], the performance of two well-known DL models (AlexNet and GoogLeNet) was compared, which were trained from transfer learning and scratch techniques. Reference [16] implemented ResNet, VGG, Inception-v4, and DenseNet models by using a fine-tuning technique. Another research compared the performance of DL architectures including AlexNet, ResNet, DenseNet, SqueezeNet, Inception-v3, and VGG by training through transfer learning and scratch techniques [24].

The research in deep learning has been progressing with the passage of time by introducing various methods to achieve remarkable outcomes. For example, in [27], a random search method was proposed for tuning the hyperparameters of the neural network to reduce forecasting errors. Similarly, various recent studies proposed the optimizations algorithms to find the optimal value of hyperparameters of DL architectures [28]. Moreover, deep learning requires an optimization algorithm to update the weight parameters and reduce the losses. Therefore, various deep learning optimizers have been developed by the research community to achieve better results in image classification tasks. These optimizers produce a significant improvement in the performance of DL models. In the context of plant disease classification, the previous researches either focused on the modification of state-of-the-art DL models or the deployment of various training techniques. However, none of the previous studies has proposed an improvement in the plant disease classification by state-of-the-art DL optimizers

through a comparative study. In this regard, this article presents a comprehensive comparative analysis to perform plant disease classification in two steps. In the first step, the performance of 18 convolutional neural networks was evaluated: 10 famous/well-known DL architectures that were previously used for several image recognition tasks, six recently published modified versions that were derived from the famous DL models, and two cascaded/hybrid versions that were developed from two efficient DL algorithms; the second step was applied to improve the performance of the best-obtained model by training with various deep learning optimizers including RMSProp, Adam, Adadelta, Adamax, and Adagrad. For a comprehensive evaluation, validation accuracy/loss, F1-score, and the number of epochs (required to converge training and validation plots) were compared. The PlantVillage dataset was selected for this research, which contains disease in 14 different plant species. The successful/better classification results obtained in a large variety of dataset classes confirm that the method presented in this article can also be applied to other datasets related to plant disease. Furthermore, the better results obtained by this research will be useful for future studies regarding the real-time classification and detection of plant disease in a single framework. Moreover, the proposed methodology could also be adopted to other agricultural applications.

The rest of the paper is organized as follows: Section 2 presents the details of the dataset, hardware/software specifications, DL architectures, DL optimizers, and specifications required to train the DL models. Section 3 presents the results to indicate the performance of all the well-known, modified, and cascaded/hybrid versions of DL models along with the improvement in the performance of best-obtained models by using various deep learning optimizers, and finally, Section 4 describes the concluding remarks along with some future recommendations.

2. Materials and Methods

The Convolutional Neural Networks (CNNs) are mostly used for image classification tasks. Therefore, in this research, the performance of many state-of-the-art CNN architectures was evaluated for the classification of plant diseases. The modified and cascaded versions of DL architectures were also considered, which were recently published in prominent research articles related to plant disease classification. Figure 1 shows all the 18 DL architectures considered for this research. These models were divided into three categories: well-known, modified/improved, and cascaded/hybrid versions. An overall methodology of this research is presented in Figure 2. Firstly, the Stochastic Gradient Descent (SGD) with momentum optimizer was selected to train the CNN models due to its fast convergence ability [24]. Then, 18 CNN architectures were trained on the PlantVillage dataset and their convergence to the final training/validation values was observed to update the hyperparameters. Next, the CNN models were compared in terms of training and validation accuracy/loss, and F1-score. This led us to apply the DL optimization algorithms for further improvement in the performance of those CNN architectures, which achieved the highest F1-score in their particular category. The novelty of the work is proved by getting the most suitable combination of the CNN model and DL optimizer, which provided considerably better result as compared to the previous researches.

2.1. Dataset

All the DL models were trained on a publicly available dataset called PlantVillage [29], which contains a total of 54,306 images containing 38 different healthy/diseased leaves related to their 14 plant species (some of the plant diseases are shown in Figure 3). The size of the images was changed to $224 \times 224 \times 3$ and normalization was considered by dividing the values of pixel by 255 for making it suitable for the initial values of the models. The dataset was divided by 70%, 20%, and 10% into three categories to avoid overfitting: training, validation, and testing datasets, respectively [22].

Plants **2020**, *9*, 1319

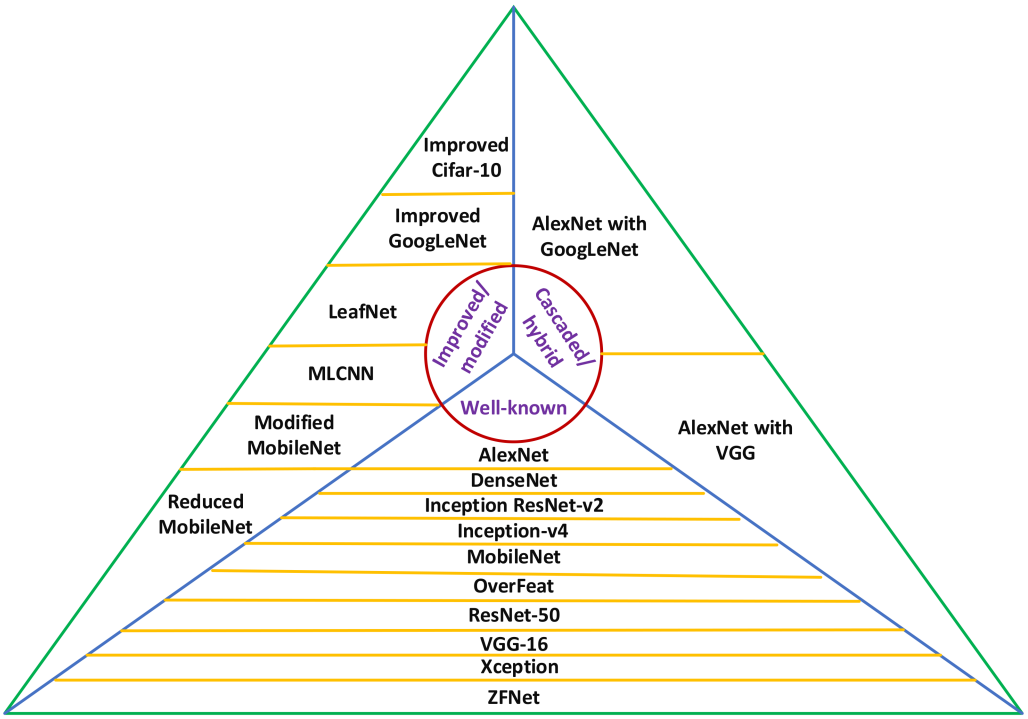


Figure 1. Three categories of DL architectures: well-known, improved/modified, and cascaded/hybrid versions. MLCNN: Multi-label Convolutional Neural Network, VGG: Visual Geometry Group.

2.2. Software and Hardware Specifications

The DL architectures were programmed in Python language due to the availability of very useful libraries and DL frameworks. Keras with TensorFlow backend was utilized to build the architectures. CuDNN library was installed as it increases the speed of training and works with TensorFlow. All the experiments were carried out on a Graphical Processing Unit (NVIDIA Quadro K2200) having the specifications: 4GB memory, 640 CUDA cores, 1045 MHz core clock, and 80 GB/sec memory bandwidth.

2.3. Deep Learning Architectures

After the development of the AlexNet architecture, a revolutionary period of state-of-the-art CNN architectures was started for many image classification tasks. Therefore, in this article, we considered very popular and successful CNN models such as AlexNet [30], OverFeat [31], VGG-16 [32], ZFNet [33], ResNet-50 [34], Inception ResNet-v2 [35], Inception-v4 [35], MobileNet [36], DenseNet-121 [37] and Xception [38].

Some researchers proposed improved/modified versions of state-of-the-art DL architectures to achieve better/more results for classifying the diseases of plant species. Among them, we have considered improved GoogLeNet [20], inspired by the famous GoogLeNet model [39], Cifar-10 [20], LeafNet [23], a multilayer convolutional neural network (MLCNN) [17] derived from the AlexNet model [30], and modified and reduced MobileNet [22] inspired by the MobileNet model [36]. Some cascaded/hybrid versions of DL architectures have also been considered in this article such as a cascaded form of the well-known AlexNet with GoogLeNet models as described in [18] and a hybrid DL architecture of AlexNet with VGG models (AgroAVNET) as proposed in [40].

Plants **2020**, *9*, 1319

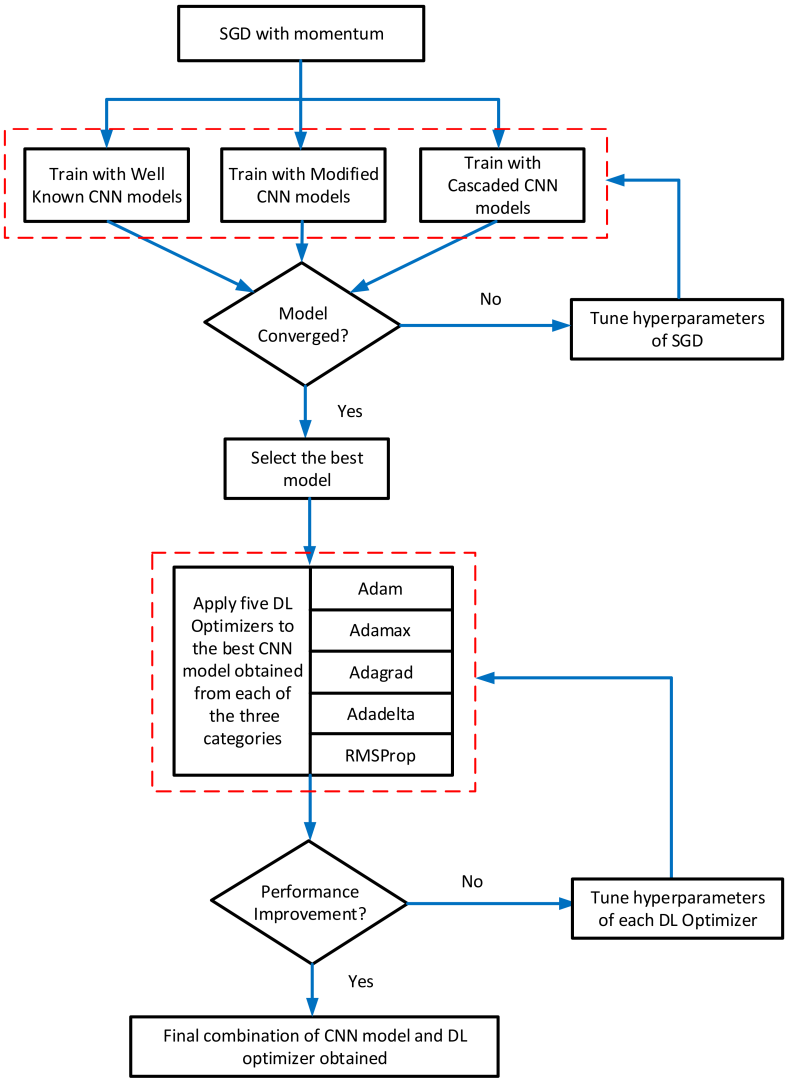


Figure 2. The methodology of this research. CNN: Convolutional Neural Network, SGD: Stochastic Gradient Descent.

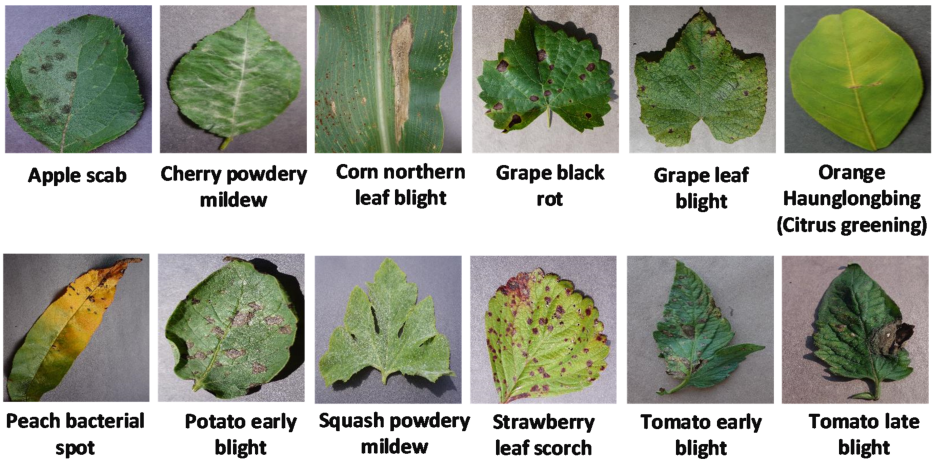


Figure 3. Some of the plant diseases from the PlantVillage dataset [29].

Plants **2020**, *9*, 1319

2.4. Deep Learning Optimizers

The Stochastic Gradient Descent (SGD) was used to train all the DL models during the first step of the proposed method. After getting the best DL architecture, an improvement in the classification of plant disease was also attempted. In this regard, we used five state-of-the-art deep learning optimizers to train those DL models which attained the highest validation accuracy and F1-score in the first step of the analysis. Few characteristics of these optimizers are provided as under:

- SGD: This is one of the simplest deep learning optimizers. A static learning rate for all the parameters requires in the duration of whole training and it has a fast convergence ability [41].
- Adagrad: This optimizer uses different learning rates for every parameter in the model. It updates the learning rate according to the frequency of the update of each parameter [42].
- RMSProp: To reduce the training time observed in Adagrad, the RMSProp optimizing functions were proposed and its learning rate decays exponentially [43].
- Adadelta: This is an extended version of Adagrad optimizer and accumulates the previous gradients over a fixed time window which ultimately ensures the continuation of learning even after many iterations. Adadelta used Hessian approximation to ensure the update direction in the negative gradient and eliminated the learning rate from update rule [44].
- Adam: The Adaptive moment estimation method (Adam) evaluates adaptive learning rates from the first and second moments of gradients for various parameters [45]. It has combined advantages of two extended versions of the SGD method that are Adagrad and RMSProp. In contrast with the RMSProp, it calculates the average of the second moment of gradient and it also utilizes the previous gradients to speed up learning [45].
- Adamax: A different version of Adam was also proposed in [45] which is based on the infinity norm and could be useful for sparse parameter updates like word embeddings.

2.5. Training Specifications

All the DL models were trained from scratch on the PlantVillage dataset. The hyperparameters were tuned by the random search method [46]. The internal covariate shift problem occurs on the neural network because of the variation in the distribution of input data due to a change in the number of parameters in the previous layer. This problem was addressed by Batch Normalization which is a very useful technique for a high learning rate [47]. For training all the DL models, the ReLU activation function was used as it is computationally efficient [24,30] and reduces the possibility of the gradient vanishing. The specifications of all the DL optimizers are summarized in Table 1.

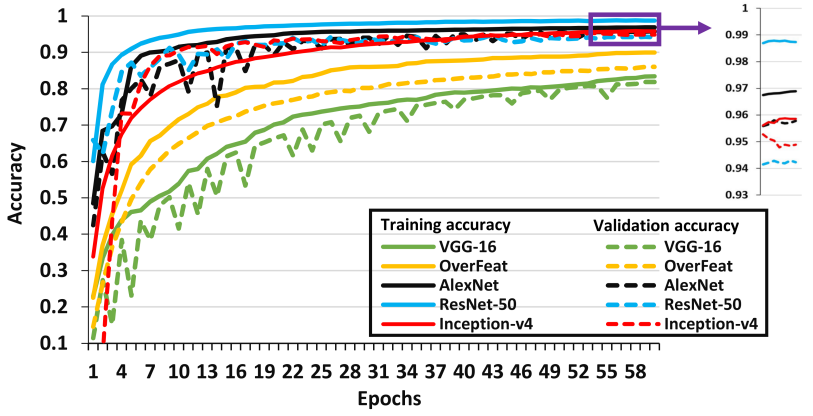
Table 1. Hyperparameters of the deep learning optimizers.

Optimizers	Specifications
SGD	learning rate = 0.001, weight decay = 0.0005, momentum = 0.9, nesterov = False
Adagrad	learning rate = 0.001, epsilon = 1×10^{-7}
RMSProp	learning rate = 0.001, rho = 0.9, epsilon = 1×10^{-7}
Adadelta	learning rate = 1.0, rho=0.95, epsilon = 1×10^{-6}
Adam	learning rate = 0.001, beta1 = 0.9, beta2 = 0.999, epsilon = 1×10^{-8}, amsgrad = False
Adamax	learning rate = 0.002, beta1 = 0.9, beta2 = 0.999, epsilon = 1×10^{-8}

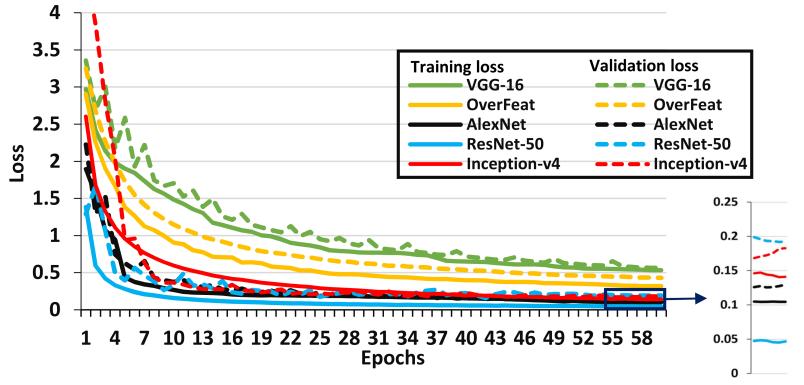
3. Results and Discussion

This section first presents the comparative analysis of DL architectures to select the best model which leads to the results obtained regarding the improvement in the performance of the best-suited

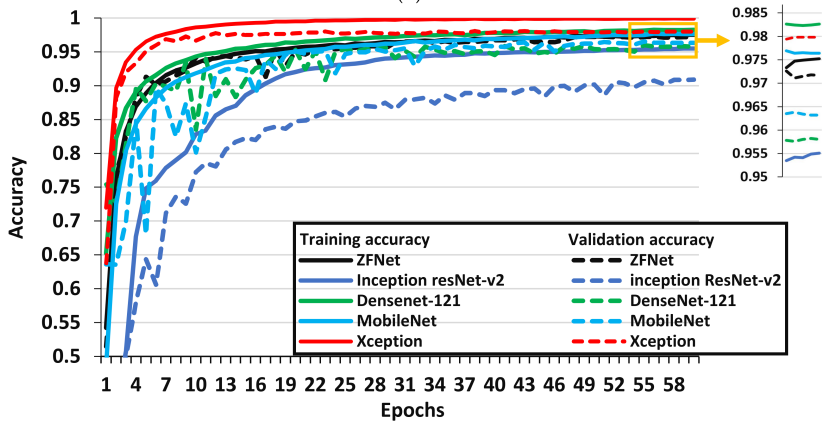
models by using various DL optimization algorithms. All the results were evaluated in terms of training, validation accuracy/loss, and F1-score. The F1-score is considered an important performance metric especially for the case when there is an uneven distribution in the classes just such as the PlantVillage dataset (for example, the Potato healthy class contains the least number of images (152), whereas, the Citrus greening has the highest number of images (5507) [29]). Therefore, the model/optimizer that attained the highest F1-score was considered the most suitable architecture for the classification of plant disease. The performances of all DL architectures are represented by line graphs (Figures 4–6), and it was empirically observed that they required 60 epochs (an epoch is a complete cycle of training on each image sample in the training dataset) at which training/validation accuracy and loss were converged. The overall performance of DL architectures is also summarized in Table 2.



(a)



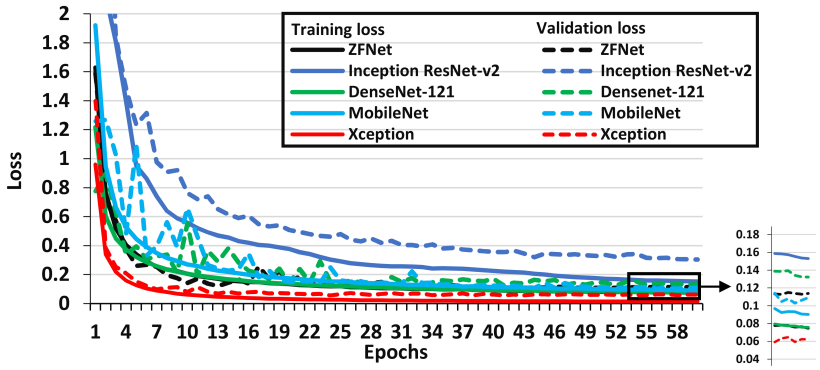
(b)



(c)

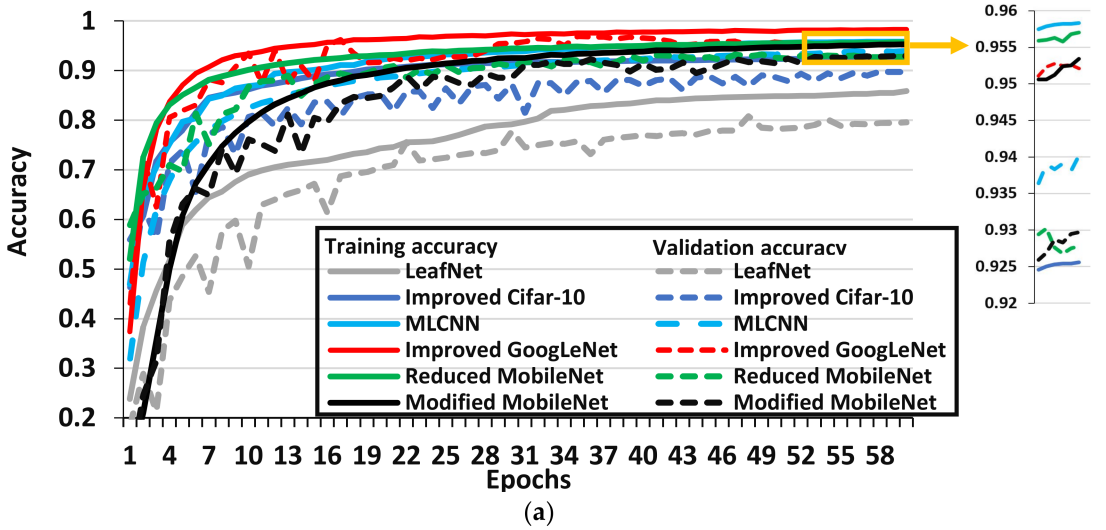
Figure 4. *Cont.*

Plants **2020**, *9*, 1319

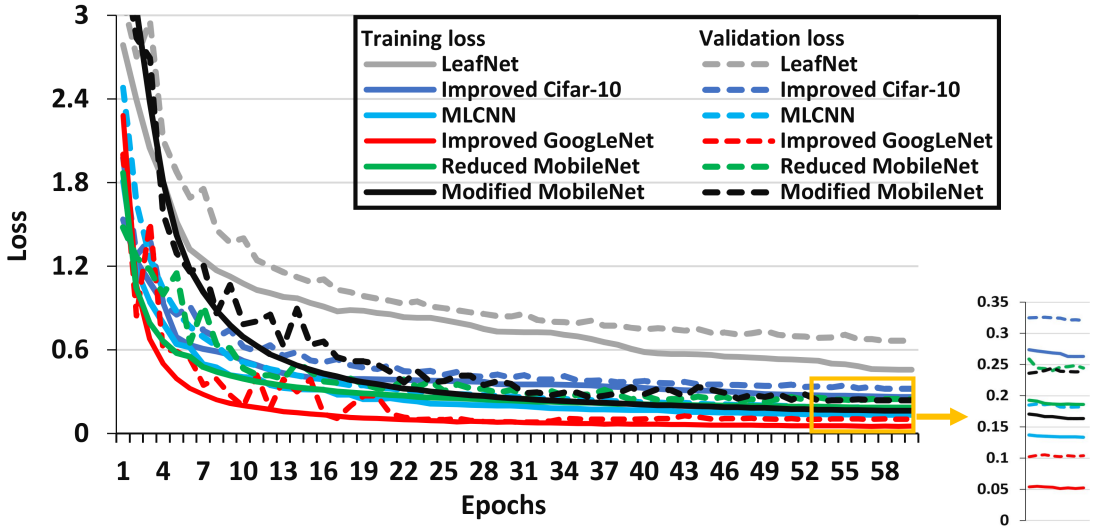


(d)

Figure 4. Performance plots of well-known CNN architectures. (**a**), (**b**) provide training and validation accuracy/loss of VGG-16, OverFeat, AlexNet, ResNet-50 and Inception-v4 architectures. (**c**), (**d**) provide training and validation accuracy/loss of ZFNet, Inception ResNet-v2, DenseNet-121, MobileNet and Xception architectures.



(a)



(b)

Figure 5. Performance plots of modified versions of CNN architectures. (**a**) Provides training and validation accuracy, (**b**) provides training and validation loss.

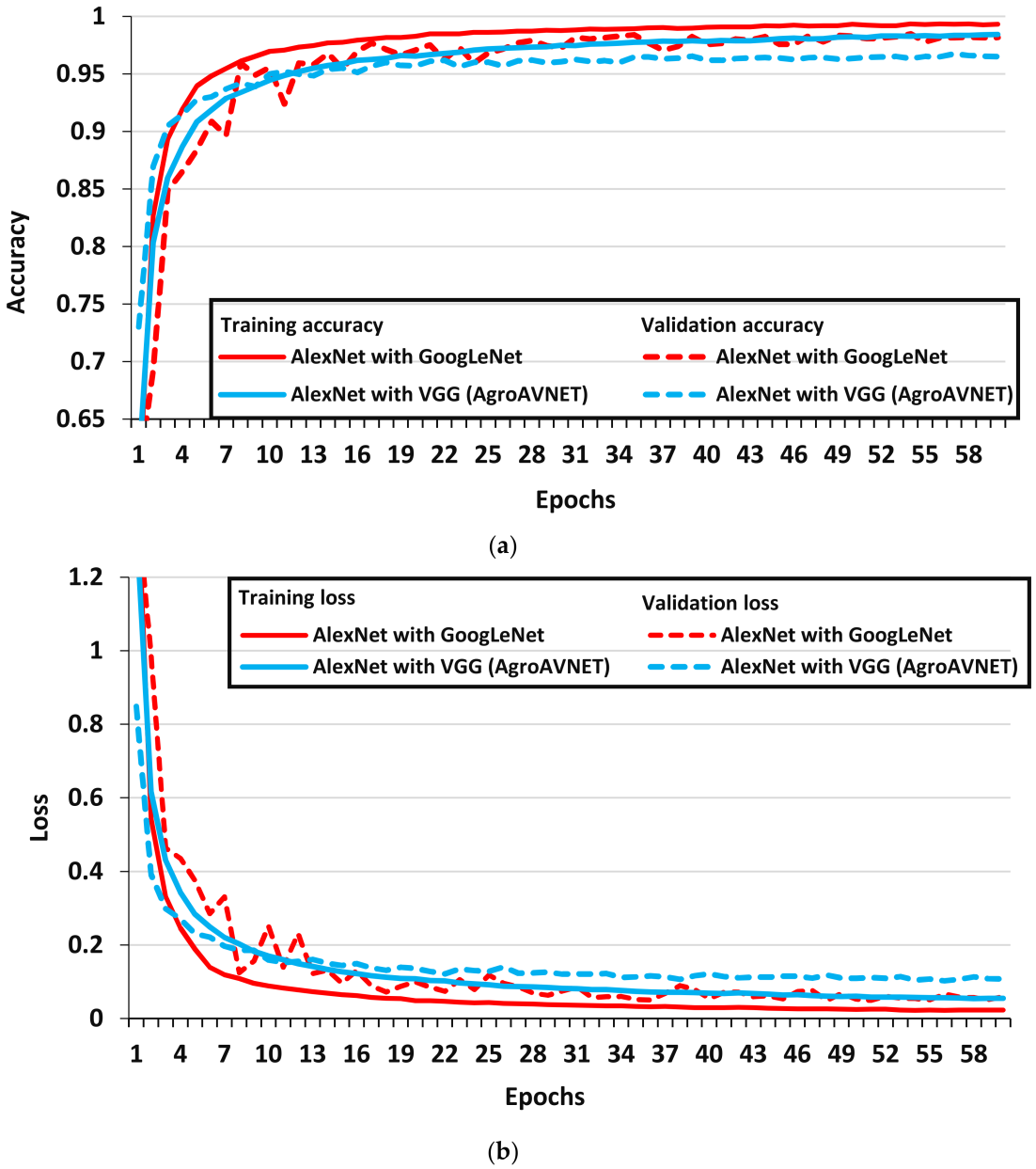


Figure 6. Performance plots of cascaded/hybrid versions of CNN architectures. (**a**) Provides training and validation accuracy, (**b**) provides training and validation loss.

Table 2. Training and validation accuracy/loss, precision, recall, and F1-score along with the number of parameters, training time, and epochs required to train deep learning architectures (in the order of the lowest to the highest F1-score).

Deep Learning Architectures	Parameters (in Millions)	Epochs Required to Train the Model	Training Time (in Hours)	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss	Precision	Recall	F1-score
LeafNet	0.324 M	59	5.95	0.8590	0.7961	0.4563	0.6658	0.7946	0.7971	0.7958
VGG-16	138 M	59	38.13	0.8339	0.8189	0.5328	0.5651	0.8182	0.8194	0.8188
OverFeat	141.8 M	58	6.75	0.8995	0.8603	0.3201	0.4330	0.8592	0.8628	0.8610
Improved Cifar-10	2.43 M	58	6.08	0.9256	0.8974	0.2628	0.3205	0.8944	0.8960	0.8952
Inception ResNet v2	54.3 M	58	32.83	0.9551	0.9091	0.1530	0.3047	0.9075	0.9105	0.9089
Reduced MobileNet	0.5 M	55	11.72	0.9570	0.9278	0.1860	0.2442	0.9269	0.9267	0.9268
Modified MobileNet	0.5 M	53	6.38	0.9534	0.9297	0.1632	0.2385	0.9278	0.9265	0.9271
ResNet-50	23.6 M	55	26.33	0.9873	0.9423	0.0468	0.1923	0.9351	0.9358	0.9354
MLCNN	78 M	57	67.33	0.9583	0.9402	0.1335	0.1820	0.9386	0.9411	0.9398
Inception v4	41.2 M	59	52.92	0.9586	0.9489	0.1410	0.1828	0.9410	0.9466	0.9438
Improved GoogLeNet	6.8 M	53	9.67	0.9829	0.9521	0.0522	0.1038	0.9528	0.9539	0.9533
AlexNet	60 M	54	6.10	0.9689	0.9578	0.1046	0.1298	0.9563	0.9570	0.9566
DenseNet-121	7.1 M	56	28.75	0.9826	0.9580	0.0758	0.1323	0.9581	0.9569	0.9575
MobileNet	3.2 M	47	14.70	0.9764	0.9632	0.0903	0.1090	0.9624	0.9612	0.9618
Hybrid AlexNet with VGG (AgroAVNET)	238 M	54	49.90	0.9841	0.9649	0.0546	0.1078	0.9626	0.9674	0.9650
ZFNet	58.5 M	47	6.47	0.9752	0.9717	0.0746	0.1139	0.9746	0.9751	0.9748
Cascaded AlexNet and GoogLeNet	5.6 M	57	6.5	0.9931	0.9818	0.0229	0.0592	0.9749	0.9751	0.9750
Xception	22.8 M	34	56.28	0.9990	0.9798	0.0140	0.0621	0.9764	0.9767	0.9765

Plants **2020**, *9*, 1319

3.1. Step-1: Comparative Analysis of Deep Learning Architectures

3.1.1. Performance of Well-Known CNN Architectures

The performance of well-known CNN architectures is presented in Figure 4, and it indicates that there is no sign of underfitting (the problem occurs during the training of deep learning models according to which the model does not train accurately if training loss does not change or it continuously decreases) and overfitting (the problem at which the model does not perform appropriately for new data/validation dataset or validation loss decreases to some extent then suddenly increases for the remaining epochs). Overall, 10 well-known CNN architectures were considered. A few important observations from Figure 4 and Table 2 were made:

- The Xception model attained the highest validation accuracy, F1-score, and lowest validation loss among all the well-known CNN models. Therefore, this model can be undoubtedly considered as the best CNN architecture to classify plant disease on the PlantVillage dataset. It implies that the concept of a modified version of depth-wise separable convolution [38] in the Xception model is a useful way to obtain higher classification results. Moreover, this DL model converged to its final value at the 34th epoch which is the least number of epochs as compare to all the other DL architectures. On the other hand, it required a significant amount of time to complete one epoch (around 3400 s). Therefore, future studies should propose another version of DL architecture that can achieve Xception-level accuracy and require smaller training time for each epoch.
- The second highest F1-score/validation accuracy was attained by ZFNet architecture. Hence, a smaller filter size and the increased number of activation maps used in ZFNet architectures (as compared to AlexNet) improved its performance.
- Then, MobileNet, DenseNet, and AlexNet architectures have also achieved a good F1-score followed by Inception-v4, ResNet-50, and Inception ResNet-v2 architectures. The MobileNet is a comparatively more preferable model due to its lower number of parameters which reduced its computation time significantly. The depthwise and pointwise convolutional layers helped to achieve a better classification result. Therefore, a CNN model could be proposed in future research based on the MobileNet architecture. Moreover, this model required a lower number of epochs to achieve its final accuracy and loss as compare to DenseNet and AlexNet models (as shown in Table 2).
- From Table 2, it is also noticed that the DL models, such as Inception-v4, Inception ResNet-v2, OverFeat, and VGG-16, required 58-59 number of epochs to converge training/validation plots (also shown in Figure 4), which significantly increased their training time.
- The VGG-16 and OverFeat were found unsuitable models for plant disease classification as they achieved lower validation accuracy/F1-score and higher validation loss as compared to the other well-known DL architectures. The smaller filter size of the VGG model degraded its performance. However, the larger filter size of the OverFeat model significantly reduced its training time but they were not enough to provide a noticeable classification performance. Additionally, they had a higher number of parameters (in millions) which slow down their training time effectively.

3.1.2. Performance of Modified CNN Architectures

In this article, six modified/improved versions of CNN architectures were also considered. Their performance is presented in Figure 5 from which the following points are discussed:

- The improved GoogLeNet architecture achieved the best performance in terms of validation accuracy/loss and F1-score among all the modified versions of CNN architectures by utilizing the concept of the Inception module from the original GoogLeNet model. Moreover, it got the final value of accuracy and loss in 53 epochs which is the least as compared to other modified/improved versions of the DL models considered in this article, but it required more training time to complete one epoch as compared to the models like Modified and Reduced MobileNet.

- The MLCNN architecture provided a good F1-score due to the inclusion of a dropout layer after each max pooling layer and a reduction in the number of filters of the starting convolution layers in the original AlexNet architecture. However, due to a higher number of parameters, this modified DL architecture required considerably higher training time per epoch.
- The two versions of MobileNet named Modified and Reduced MobileNet models achieved an acceptable F1-score closed to each other. These modified versions of DL architecture used depthwise separable convolutional layers, which helped to attain a good classification result, and they had six times fewer parameters than the original MobileNet model which reduced their training time per epoch.
- Moreover, there were some models like Improved Cifar-10 and LeafNet models that had a lower number of parameters which increased their speed of training per epoch. The Improved Cifar-10 model achieved a noticeable F1-score, but the reduced parameters of the LeafNet model were not enough to obtain a good F1-score/validation accuracy. Therefore, it is not a suitable model to classify diseases in the selected dataset. It is also observed that these two models required a higher number of epochs as compare to other modified versions of DL architectures. Hence, future research could comprise of proposing a DL model such as Improved Cifar-10 and LeafNet for reducing the training time, but some convolutional layers should be added to attain acceptable validation/testing accuracy.

3.1.3. Performance of Cascaded/Hybrid CNN Architectures

Figure 6 presents the performance of cascaded/hybrid version of CNN models as explained below:

- The cascaded AlexNet with GoogLeNet architecture outperformed all the DL models in terms of validation accuracy; moreover, except for the Xception architecture, this model achieved the highest F1-score among all the DL architectures considered in this research (as shown in Table 2). Although it required almost 57 epochs to reach its final accuracy/loss values (as shown in Figure 6), but it completed one epoch in a smaller period, which clearly shows its effectiveness in terms of training time. There were a few important modifications in the original AlexNet model, which helped to extract the features of plants containing disease including smaller convolution kernel in different layers, the inclusion of max-pooling layer, cascading the Inception module with the modified AlexNet layers, and convolutional layers after Inception to replace two fully connected layers [18].
- Moreover, a hybrid version of AlexNet with VGG architectures has also been studied, and it provided good performance in terms of validation accuracy (as shown in Figure 6) and F1-score, but it had the highest number of parameters which significantly increased its training time to complete each epoch. This model performed well due to the utilization of concepts such as normalization and selection of filter depth from AlexNet and VGG models, respectively [40].

3.2. Step-2: Improvement in Classification Results by Deep Learning Optimizers

In this article, an improvement in the performance of CNN architectures has also been attempted by training the best models (obtained from the previous step) through different deep learning optimization functions. In this regard, the best DL model was selected from each of the three categories such as the Xception, Improved GoogLeNet, and cascaded version of AlexNet with GoogLeNet models. Table 3 summarizes the results obtained by using various optimization algorithms. Some important observations can be made as follows:

- Considerable changes were observed in training/validation accuracy, loss, precision, recall, and F1-score by training the DL models through various deep learning optimizers.
- Adam and Adadelta were the most successful optimizers for all the three selected DL architectures.
- The Xception model trained with the Adam optimizer achieved the highest validation accuracy and F1-score of 99.81% and 0.9978, respectively, which clearly show the effectiveness of the

proposed approach. Moreover, these results are better than previous studies that used the same dataset but different approaches [12,16,19,24]. Therefore, the methodology proposed in this article could be used for various other agricultural operations.

- The cascaded AlexNet with GoogLeNet and improved GoogLeNet models achieved their best classification results by using the Adadelta and Adam optimizers, respectively.
- However, a degradation in the performance has also been observed when optimizing functions were changed from SGD to Adagrad and RMSProp for Xception and cascaded models, respectively.
- It is also noticed that the Improved GoogLeNet showed its lowest validation accuracy/F1-score when it was trained by the SGD optimizer.

Table 3. Performance of deep learning optimizers applied to train cascaded AlexNet with GoogLeNet, Improved GoogLeNet, and Xception models.

Optimizers	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss	Precision	Recall	F1-score
Cascaded AlexNet with GoogLeNet							
SGD	0.9931	0.9818	0.0229	0.0592	0.9749	0.9751	0.9750
RMSProp	0.9894	0.9757	0.0482	0.1479	0.9746	0.9613	0.9679
Adagrad	0.9956	0.9824	0.0153	0.0547	0.9815	0.9782	0.9798
Adamax	0.9990	0.9859	0.0029	0.0574	0.9828	0.9795	0.9811
Adam	0.9989	0.9857	0.0039	0.0750	0.9836	0.9836	0.9836
Adadelta	0.9993	0.9873	0.0024	0.0696	0.9846	0.9856	0.9851
Improved GoogLeNet							
SGD	0.9829	0.9521	0.0522	0.1038	0.9528	0.9539	0.9533
RMSProp	0.9723	0.9685	0.1780	0.2272	0.9692	0.9666	0.9679
Adagrad	0.9889	0.9718	0.0350	0.0930	0.9651	0.9618	0.9634
Adamax	0.9998	0.9847	8.782×10^{-4}	0.0875	0.9792	0.9826	0.9809
Adam	0.9992	0.9904	0.0026	0.0434	0.9859	0.9872	0.9864
Adadelta	0.9991	0.9905	0.0022	0.0567	0.9828	0.9879	0.9861
Xception							
SGD	0.9990	0.9798	0.0140	0.0621	0.9764	0.9767	0.9765
RMSProp	0.9998	0.9924	6.922×10^{-4}	0.0433	0.9877	0.9920	0.9900
Adagrad	0.9987	0.9621	0.0164	0.1460	0.9682	0.9505	0.9593
Adamax	1.0000	0.9889	0.0012	0.0415	0.9902	0.9874	0.9888
Adam	1.0000	0.9981	6.890×10^{-4}	0.0178	0.9981	0.9975	0.9978
Adadelta	1.0000	0.9906	8.407×10^{-4}	0.0364	0.9926	0.9887	0.9906

4. Conclusions and Future Recommendations

In this article, a comprehensive comparative analysis has been performed between various state-of-the-art deep learning architectures divided into three categories namely well-known, modified, and cascaded versions. Moreover, the performance of the best-obtained models was further improved by using various deep learning optimization algorithms. It was found that the Xception, Improved GoogLeNet and cascaded version of AlexNet with GoogLeNet models obtained the highest validation accuracy and F1-score in their respective category. When these three DL models were trained by using various deep learning optimizers, the Xception model trained by the Adam optimizer achieved the highest F1-score of 0.9978 which suggests that this combination of the CNN model and the optimization algorithm is the most suitable way to classify the plant disease. This research provided us some interesting future directions for upcoming research given as follows:

- Various deep learning optimizers such as Adam, and Adadelta, can also be used to enhance research on other agricultural applications, such as crop/weed discrimination, classification of weeds, plant recognition, etc.
- The classification performance of the other datasets related to plant disease could also be improved by adopting the methodology proposed in this research.
- Furthermore, although the Xception model provided the best results according to the analysis provided in this article, it required a significant amount of time to complete each epoch. Therefore, an attempt should be made to achieve an Xception level accuracy with small training time.

Author Contributions: Conceptualization, M.H.S. and K.M.A.; methodology, M.H.S. and K.M.A.; investigation, M.H.S. and K.M.A.; writing—original draft preparation, M.H.S. and K.M.A.; writing—review and editing, M.H.S. and K.M.A.; visualization, M.H.S. and K.M.A.; supervision, J.P. and K.M.A.; project administration, J.P. and K.M.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of Business, Innovation and Employment (MBIE), New Zealand, Science for Technological Innovation (SfTI) National Science Challenge.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liakos, K.G.; Busato, P.; Moshou, D.; Pearson, S.; Bochtis, D. Machine learning in agriculture: A review. *Sensors* **2018**, *18*, 2674. [CrossRef] [PubMed]
2. Römer, C.; Bürling, K.; Hunsche, M.; Rumpf, T.; Noga, G.; Plümer, L. Robust fitting of fluorescence spectra for pre-symptomatic wheat leaf rust detection with support vector machines. *Comput. Electron. Agric.* **2011**, *79*, 180–188. [CrossRef]
3. Chen, T.; Zhang, J.; Chen, Y.; Wan, S.; Zhang, L. Detection of peanut leaf spots disease using canopy hyperspectral reflectance. *Comput. Electron. Agric.* **2019**, *156*, 677–683. [CrossRef]
4. Coops, N.; Stanford, M.; Old, K.; Dudzinski, M.; Culvenor, D.; Stone, C. Assessment of Dothistroma needle blight of Pinus radiata using airborne hyperspectral imagery. *Phytopathology* **2003**, *93*, 1524–1532. [CrossRef]
5. Leucker, M.; Mahlein, A.-K.; Steiner, U.; Oerke, E.-C. Improvement of lesion phenotyping in Cercospora beticola–sugar beet interaction by hyperspectral imaging. *Phytopathology* **2015**, *106*, 177–184. [CrossRef] [PubMed]
6. Saleem, M.H.; Potgieter, J.; Mahmood Arif, K. Plant Disease Detection and Classification by Deep Learning. *Plants* **2019**, *8*, 468. [CrossRef]
7. Xie, C.; Yang, C.; He, Y. Hyperspectral imaging for classification of healthy and gray mold diseased tomato leaves with different infection severities. *Comput. Electron. Agric.* **2017**, *135*, 154–162. [CrossRef]
8. Kobayashi, T.; Kanda, E.; Kitada, K.; Ishiguro, K.; Torigoe, Y. Detection of rice panicle blast with multispectral radiometer and the potential of using airborne multispectral scanners. *Phytopathology* **2001**, *91*, 316–323. [CrossRef]
9. Brahimi, M.; Boukhalfa, K.; Moussaoui, A. Deep learning for tomato diseases: Classification and symptoms visualization. *Appl. Artif. Intel.* **2017**, *31*, 299–315. [CrossRef]
10. Ferentinos, K.P. Deep learning models for plant disease detection and diagnosis. *Comput. Electron. Agric.* **2018**, *145*, 311–318. [CrossRef]
11. Fuentes, A.; Yoon, S.; Kim, S.; Park, D. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors* **2017**, *17*, 2022. [CrossRef] [PubMed]
12. Mohanty, S.P.; Hughes, D.P.; Salathé, M. Using deep learning for image-based plant disease detection. *Front. Plant Sci.* **2016**, *7*, 1419. [CrossRef] [PubMed]
13. TÜRKOĞLU, M.; Hanbay, D. Plant disease and pest detection using deep learning-based features. *Turk. J. Electr. Eng. Comput. Sci.* **2019**, *27*, 1636–1651. [CrossRef]
14. Zhang, K.; Wu, Q.; Liu, A.; Meng, X. Can Deep Learning Identify Tomato Leaf Disease? *Adv. Multimed.* **2018**, *2018*. [CrossRef]
15. Oppenheim, D.; Shani, G.; Erlich, O.; Tsror, L. Using Deep Learning for Image-Based Potato Tuber Disease Detection. *Phytopathology* **2019**, *109*, 1083–1087. [CrossRef] [PubMed]
16. Too, E.C.; Yujian, L.; Njuki, S.; Yingchun, L. A comparative study of fine-tuning deep learning models for plant disease identification. *Comput. Electron. Agric.* **2019**, *161*, 272–279. [CrossRef]

17. Singh, U.P.; Chouhan, S.S.; Jain, S.; Jain, S. Multilayer Convolution Neural Network for the Classification of Mango Leaves Infected by Anthracnose Disease. *IEEE Access* **2019**, *7*, 43721–43729. [CrossRef]
18. Liu, B.; Zhang, Y.; He, D.; Li, Y. Identification of apple leaf diseases based on deep convolutional neural networks. *Symmetry* **2018**, *10*, 11. [CrossRef]
19. Geetharamani, G.; Pandian, A. Identification of plant leaf diseases using a nine-layer deep convolutional neural network. *Comput. Electr. Eng.* **2019**, *76*, 323–338.
20. Zhang, X.; Qiao, Y.; Meng, F.; Fan, C.; Zhang, M. Identification of maize leaf diseases using improved deep convolutional neural networks. *IEEE Access* **2018**, *6*, 30370–30377. [CrossRef]
21. Chen, J.; Liu, Q.; Gao, L. Visual Tea Leaf Disease Recognition Using a Convolutional Neural Network Model. *Symmetry* **2019**, *11*, 343. [CrossRef]
22. Kamal, K.; Yin, Z.; Wu, M.; Wu, Z. Depthwise separable convolution architectures for plant disease classification. *Comput. Electron. Agric.* **2019**, *165*, 104948.
23. Jiang, P.; Chen, Y.; Liu, B.; He, D.; Liang, C. Real-Time Detection of Apple Leaf Diseases Using Deep Learning Approach Based on Improved Convolutional Neural Networks. *IEEE Access* **2019**, *7*, 59069–59080. [CrossRef]
24. Brahimi, M.; Arsenovic, M.; Laraba, S.; Sladojevic, S.; Boukhalfa, K.; Moussaoui, A. Deep learning for plant diseases: Detection and saliency map visualisation. In *Human and Machine Learning*; Springer: Berlin, Germany, 2018; pp. 93–117.
25. Brahimi, M.; Mahmoudi, S.; Boukhalfa, K.; Moussaoui, A. Deep interpretable architecture for plant diseases classification. *arXiv* **2019**, arXiv:1905.13523.
26. DeChant, C.; Wiesner-Hanks, T.; Chen, S.; Stewart, E.L.; Yosinski, J.; Gore, M.A.; Nelson, R.J.; Lipson, H. Automated identification of northern leaf blight-infected maize plants from field imagery using deep learning. *Phytopathology* **2017**, *107*, 1426–1432. [CrossRef]
27. Torres, J.F.; Gutiérrez-Avilés, D.; Troncoso, A.; Martínez-Álvarez, F. Random hyper-parameter search-based deep neural network for power consumption forecasting. In Proceedings of the International Work-Conference on Artificial Neural Networks (IWANN 2019), Gran Canaria, Spain, 12–14 June 2019; pp. 259–269.
28. Martínez-Álvarez, F.; Asencio-Cortés, G.; Torres, J.F.; Gutiérrez-Avilés, D.; Melgar-García, L.; Pérez-Chacón, R.; Rubio-Escudero, C.; Riquelme, J.; Troncoso, A. Coronavirus optimization algorithm: A bioinspired metaheuristic based on the COVID-19 propagation model. *Big Data* **2020**, *8*, 308–322. [CrossRef]
29. Hughes, D.; Salathé, M. An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv* **2015**, arXiv:1511.08060.
30. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems (NIPS 2012), Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
31. Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu, M.; Fergus, R.; LeCun, Y. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv* **2013**, arXiv:1312.6229.
32. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
33. Zeiler, M.D.; Fergus, R. Visualizing and understanding convolutional networks. In Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; pp. 818–833.
34. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
35. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17), San Francisco, CA, USA, 4–9 February 2017; pp. 4278–4284.
36. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
37. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Hawaii Convention Center, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
38. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Hawaii Convention Center, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.

39. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9.
40. Chavan, T.R.; Nandedkar, A.V. AgroAVNET for crops and weeds classification: A step forward in automatic farming. *Comput. Electron. Agric.* **2018**, *154*, 361–372. [CrossRef]
41. Ruder, S. An overview of gradient descent optimization algorithms. *arXiv* **2016**, arXiv:1609.04747.
42. Duchi, J.; Hazan, E.; Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.* **2011**, *12*, 2121–2159.
43. Hinton, G.; Srivastava, N.; Swersky, K. Neural networks for machine learning. Available online: http://www.cs.toronto.edu/~{}hinton/coursera/lecture6/lec6.pdf (accessed on 5 October 2020).
44. Zeiler, M.D. Adadelta: An adaptive learning rate method. *arXiv* **2012**, arXiv:1212.5701.
45. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
46. Bergstra, J.; Bengio, Y. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* **2012**, *13*, 281–305.
47. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv* **2015**, arXiv:1502.03167.

Chapter 4 Image-Based Plant Disease Identification by Deep Learning Meta-Architectures

This chapter contains content from the following article.

M. H. Saleem, S. Khanchi, J. Potgieter, and K. M. Arif, "Image-based plant disease identification by deep learning meta-architectures," *Plants*, vol. 9, no. 11, p. 1451, 2020. https://doi.org/10.3390/plants9111451

plants

MDPI

Article

Image-Based Plant Disease Identification by Deep Learning Meta-Architectures

Muhammad Hammad Saleem [1], Sapna Khanchi [1], Johan Potgieter [2] and Khalid Mahmood Arif [1,*]

[1] Department of Mechanical and Electrical Engineering, School of Food and Advanced Technology, Massey University, Auckland 0632, New Zealand; H.Saleem@massey.ac.nz (M.H.S.); Sapna.Sapna.1@uni.massey.ac.nz (S.K.)

[2] Massey Agritech Partnership Research Centre, School of Food and Advanced Technology, Massey University, Palmerston North 4442, New Zealand; J.Potgieter@massey.ac.nz

* Correspondence: K.Arif@massey.ac.nz

Received: 9 October 2020; Accepted: 25 October 2020; Published: 27 October 2020

Abstract: The identification of plant disease is an imperative part of crop monitoring systems. Computer vision and deep learning (DL) techniques have been proven to be state-of-the-art to address various agricultural problems. This research performed the complex tasks of localization and classification of the disease in plant leaves. In this regard, three DL meta-architectures including the Single Shot MultiBox Detector (SSD), Faster Region-based Convolutional Neural Network (RCNN), and Region-based Fully Convolutional Networks (RFCN) were applied by using the TensorFlow object detection framework. All the DL models were trained/tested on a controlled environment dataset to recognize the disease in plant species. Moreover, an improvement in the mean average precision of the best-obtained deep learning architecture was attempted through different state-of-the-art deep learning optimizers. The SSD model trained with an Adam optimizer exhibited the highest mean average precision (mAP) of 73.07%. The successful identification of 26 different types of defected and 12 types of healthy leaves in a single framework proved the novelty of the work. In the future, the proposed detection methodology can also be adopted for other agricultural applications. Moreover, the generated weights can be reused for future real-time detection of plant disease in a controlled/uncontrolled environment.

Keywords: deep learning; plant disease detection; transfer learning; optimization algorithms; mean average precision

1. Introduction

In agricultural crops, leaves play a vital role to provide information about the amount and nature of horticultural yield. Several factors affect food production such as climate change, presence of weed, and soil infertility. Apart from that, plant or leaf disease is a global threat to the growth of several agricultural products and a source of economic losses [1]. The failure to diagnose infections/bacteria/virus in plants leads subsequently to insufficient pesticide/fungicide use. Therefore, plant diseases have been largely considered in the scientific community, with a focus on the biological features of diseases. Precision farming uses the most advanced technology for the optimization of decision-making. The visual inspections by experts and biological review are usually carried out through plant diagnosis when required. This method, however, is typically time-consuming and cost ineffective. To address these issues, it is necessary to detect plant diseases by advanced and intelligent techniques.

Plants **2020**, *9*, 1451

To perform the agricultural operations, conventional machine learning (ML) algorithms have been applied in many studies [2,3]. However, recently, deep learning (DL) as a sub-set of ML, has been strikingly effective for real-life object detection, recognition, and classification purposes [4–6]. Therefore, agricultural research has been moving towards the DL-based solutions. The DL techniques have been accomplished state-of-the-art results to perform the agricultural operations including crop/weed discrimination [7,8], fruit harvesting [9,10], and plant recognition [11–14]. Similarly, recent studies have also focused on another important agricultural issue of plant disease identification [6].

Several state-of-the-art DL models have been applied to perform plant disease classification by using well-known DL architectures. Moreover, some researchers introduced modified versions of DL algorithms to improve the performance of the classification of disease in several plant species. A few of the prominent/recent studies are highlighted in this section. For example, a recent article presented acomparative analysis of various Convolutional Neural Networks (CNN) and DL optimizers to attain better results of plant disease classification [15]. A study proposed a CNN model to classify disease in tea leaves [16]. Another study was conducted to propose two revised versions of MobileNet models for the classification of several plant diseases [17]. A recent article presented two deep learning architectures based on residual learning and attention methods to classify tomato leaf diseases and achieved a higher overall accuracy [18]. Another CNN-based architecture was proposed to classify disease in the PlantVillage dataset, and it performed better than the well-known DL models including AlexNet, VGG-16, Inception-v3, and ResNet [19]. A recent article proposed a CNN-based model for the classification of groundnut disease [20]. Similarly, few studies focused on the advanced training techniques; for example, [21] evaluated the performance of AlexNet and GoogLeNet trained from scratch and transfer learning approaches. A comparative study was conducted to show the significance of the fine-tuning technique by comparing state-of-the-art DL architectures for the classification of plant disease [22]. More recent developments regarding the specific task of plant disease classification are comprehensively presented in [6,15].

To address the task of object identification, the classification and localization of objects are performed in a single platform by using deep learning meta-architectures. In this regard, few DL algorithms have been developed. The Region-based Convolution Neural Network (RCNN) was among the first modern techniques towards image detection tasks through CNN [23]. Afterward, the successful implementation of regional proposal methods proved significant developments in object identification. In the context of plant disease recognition, very few studies have been conducted toperform this complex agricultural operation by DL techniques. For example, in [24], the deep learning models were implemented to perform plant disease localization and diagnosis. The authors used their own annotated images of tomato leaf and successfully obtained a higher mean average precision. In [25], two different approaches were developed and compared to perform automated pest detection based on ML/DL learning strategies. This work focused on the detection of the harmful pest in greenhouse tomato and pepper crops. Their findings showed that the deep learning methods provided a better result as compared to the machine learning algorithms due to its capability to perform detection and classification tasks in one step. A recent article presented the DL approach to diagnose disease in Cassava leaves by using the Single Shot MultiBox Detector (SSD) and achieved satisfactory results [26]. Another recent research considered the plant disease recognition task by CNN to estimate the severity of defects in the plant leaves [27].

From the literature, it can be concluded that most of the recent researches have been focused on the task of plant disease classification (only classify the type of disease among several plant species). However, the complex task of plant disease identification (both localization and classification of the disease in the plant) has been given very little attention. Moreover, none of the previous approaches has performed a comprehensive study regarding the detection/identification of 38 classes of plant disease by advanced DL meta-architectures. Therefore, in this research, an evaluation of three successful DL-based object detection techniques including the Single Feed-forward Neural Network, Region Proposal Network, and Region-based Fully Convolutional Network has been carried out using a transfer

learning technique that focused on an important agricultural problem of plant disease identification. The transfer learning technique is applied due to its successful performance for many object recognition tasks. From the practical point of view, reuse or transfer of information from previously learned tasks for learning new tasks increases the accuracy of the DL architectures. In this research, we have shown the final ConvNet checkpoints of the detection tools. Moreover, recently, the research community is focusing on better optimization of weight parameters of neural networks [15]. Thereby, in this work, the performance of three state-of-the-art deep learning optimizers was also analysed, which significantly improved the prediction ability (true positive detection rate) of top selected DL meta-architectures.

The main contributions of this research are summarized as follows:

1. A comprehensive study of deep learning meta-architectures has been conducted for the identification of disease in several plant species infected by fungi, infection, virus, and bacteria.
2. An attempt has been made towards the improvement in the performance of DL meta-architectures specifically for plant disease recognition/identification tasks by using three different state-of-the-art DL optimization methods including Stochastic Gradient Descent (SGD) with Momentum, Adaptative Moment Estimation (Adam), and Root Mean Square Propagation (RMSProp).
3. The weights obtained after the training of the DL models could also be used for the other datasets related to plant disease.

The rest of the article is presented as follows: Section 2 explains the overall methodology, applied framework, selection of datasets, annotation of dataset images, DL meta-architecture, DL optimizers, experimental setup, and performance metric. Section 3 presents the performance of all the DL methods along with the improvement in their performance by optimization algorithms, and Section 4 provides the conclusion with some future works.

2. Materials and Methods

This article addresses the plant disease identification task by state-of-the-art three deep learning meta-architectures prominently Faster Region-based Convolutional Neural Network (RCNN), Single Shot MultiBox Detector (SSD), and Region-based Fully Convolutional Networks (RFCN). The overall methodology for this research is presented in Figure 1. The first step was the selection of two datasets: a large dataset to obtain the pre-trained weights for transfer learning, and the second dataset was related to the different classes of disease on plant leaves. The next step was the annotation of the training dataset by an online available tool called LabelImg. This led to constructing and training the DL architectures. Then, the recognition of all the classes of plant disease was checked to tune the hyperparameters of the SGD optimizer. Next, the performance of the learned neural networks was evaluated on the images of the testing (unseen) dataset. Here, the actual outputs were compared with the expected outputs to identify errors. Furthermore, the mean average precision (mAP) of all the neural networks was measured to obtain the best suited DL model. The further improvement in the mAP was proposed by using various DL optimizers. Finally, the proposed method was successful to classify and localize the healthy/diseased leaves of various plant species.

Plants **2020**, *9*, 1451

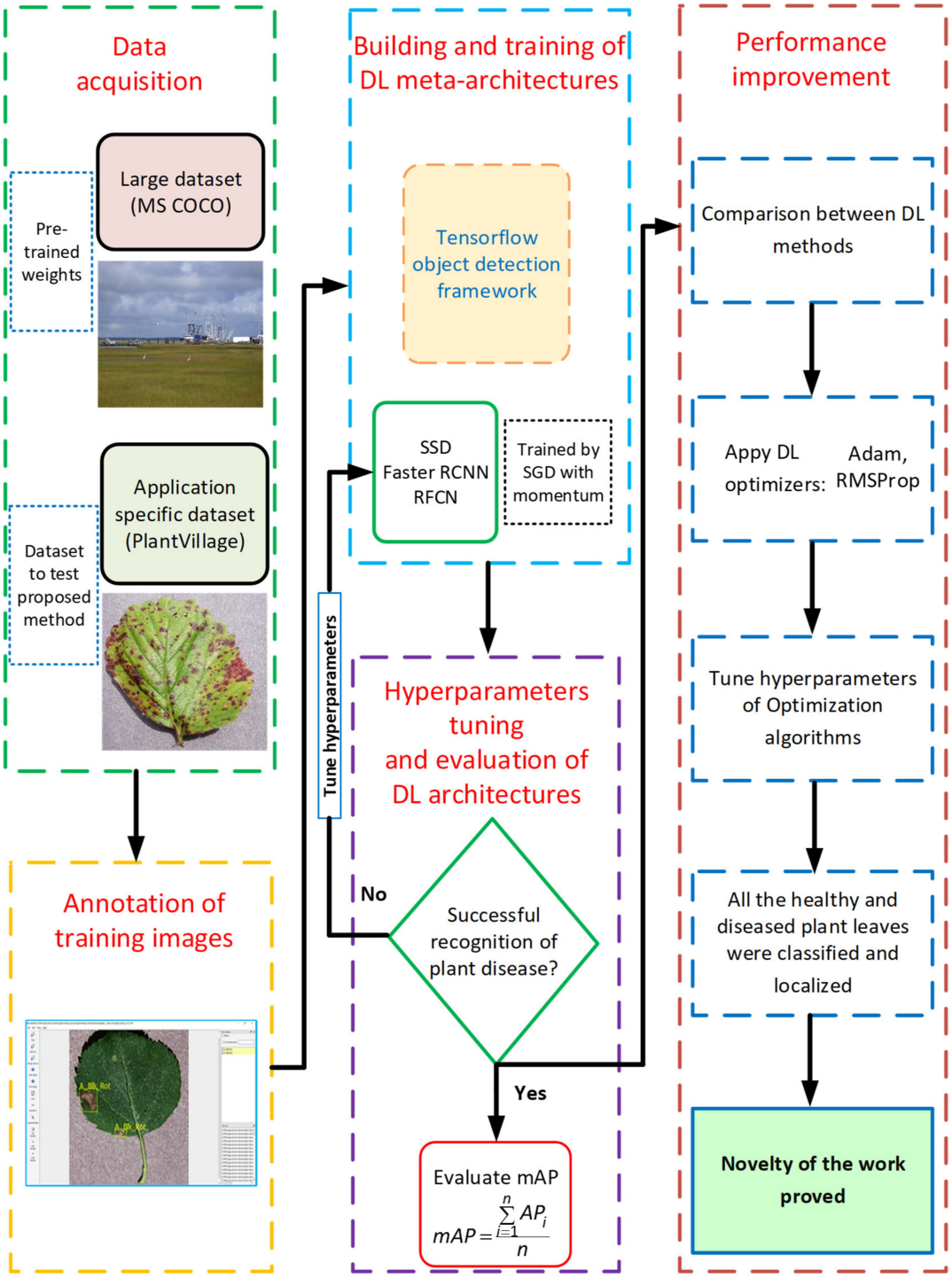


Figure 1. The flow diagram of this research.

2.1. Generalized Framework

The generalized framework of training and testing the DL models is presented in Figure 2 which consists of dataset images having their corresponding XML files. The XML data were converted into CSV format. Then, TF records from the CSV files were generated, as TensorFlow accepts the TF format of the data to feed into the network while training the DL architectures. The DL detectors were constructed by taking training images with bounding box coordinates and then evaluated their performance on the testing dataset.

Plants **2020**, *9*, 1451

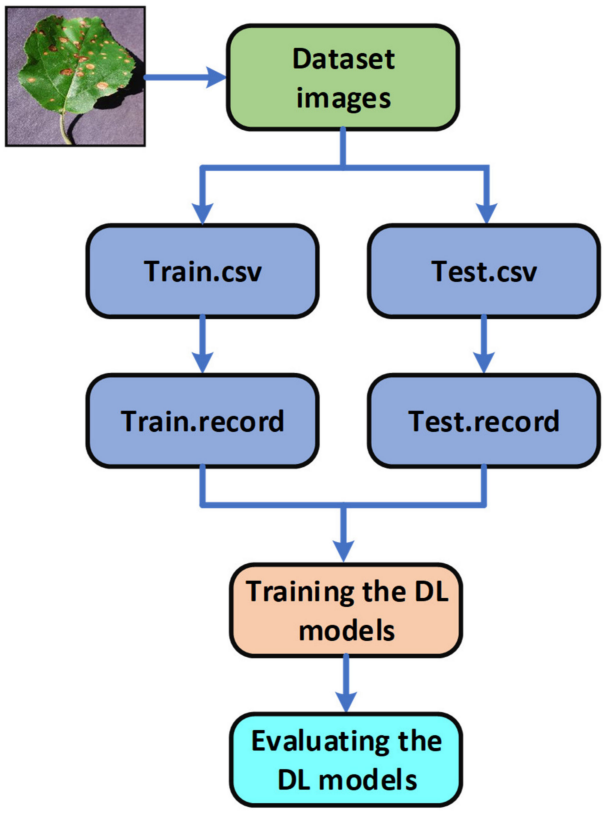


Figure 2. Generalize framework to train and test the deep learning (DL) meta-architectures for plant disease identification.

2.2. Dataset Selection

Few datasets have been developed and used for various real-life operations consisting of a huge number of classes. For example, in object classification/detection research, the dataset of ImageNet [28], which includes unprecedented numbers of images, has recently made breakthroughs. Similarly, the MS COCO dataset [29] consists of 91 common object classes with 82 of these having more than 5k labelled instances. A total of 2500k data instances are labelled in 328k pictures. The MS COCO dataset contains substantially more object instances per picture (7.7) as compared to the ImageNet (3.0) and PASCAL (2.3) datasets. Therefore, we used the training weights of the MS COCO dataset for the transfer learning purpose. Next, the PlantVillage dataset [30] was selected, as it contains images that are relevant to the area of interest. This dataset consists of images of 14 plant varieties. The dataset shows 17 fungal infections, 4 bacterial diseases, 2 fungal illnesses, 2 infectious diseases, and 1 mite-induced disease [30]. Twelve plant species also show images of healthy leaves that have no obvious illness.

2.3. Annotation of the Training Dataset

The PlantVillage dataset was divided into three sub-datasets: 70% (38017 images) for training, 20% (10858 images) for validation, and 10% (5431 images) for testing [15,17]. Then, the annotation of training dataset images was the first step towards the plant disease identification task by DL meta-architectures. In this study, the training images were annotated by LabelImg, which is an open-source graphic image annotation application. As a result, the bounding box coordinates (*Xmin*, *Ymin*, *Xmax*, and *Ymax*) were created. These bounding boxes are the ground truth boxes that evaluate as the intersection of the union (IoU) with the prediction bounding box. To save annotations as XML files, the Pascal VOC format was used. An example of an annotated dataset image is given in Figure 3. Table 1 shows the details regarding the classes of the PlantVillage dataset.

Plants **2020**, *9*, 1451

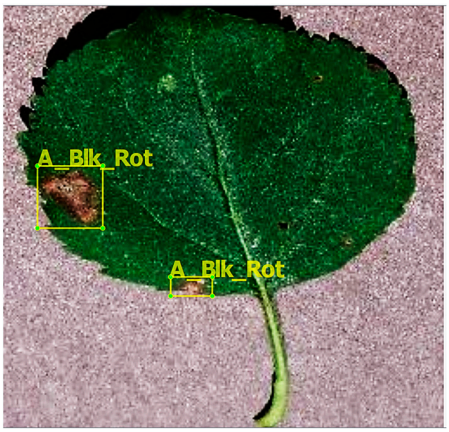


Figure 3. An example of an annotated image using the LabelImg tool.

Table 1. List of classes of PlantVillage dataset along with the cause of disease, annotation labels, number of training, validation, and testing images.

Classes of PlantVillage Dataset	Disease Cause	Annotation Label	Training Images	Validation Images	Testing Images
Apple Scab	Fungi	A_Scab	441	126	63
Apple Black Rot	Fungi	A_Blk_Rot	435	124	62
Apple Cedar Rust	Fungi	A_C_Rust	192	55	28
Apple Healthy	-	A_Healthy	1151	329	165
Blueberry Healthy	-	B_Healthy	1051	300	151
Cherry Healthy	-	Ch_Healthy	598	171	85
Cherry Powdery Mildew	Fungi	Ch_Mildew	736	210	106
Corn (maize) Common rust	Fungi	Corn_Rust	835	238	119
Corn (maize) Healthy	-	Corn_Healthy	813	233	116
Corn (maize) Northern Leaf Blight	Fungi	Corn_Blight	690	197	98
Corn (maize) Gray leaf spot	Fungi	Corn_Spot	360	102	52
Grape Black Rot	Fungi	G_Blk_Rot	826	236	118
Grape (Black Measles)	Fungi	G_Blk_Measles	968	277	138
Grape Healthy	-	Grp_Healthy	296	85	42
Grape Leaf Blight (Isariopsis Leaf Spot)	Fungi	Grp_Blight	753	215	108
Orange Huanglongbing (Citrus greening)	Bacteria	O_HLBing	3855	1101	551
Peach Bacterial Spot	Bacteria	Pec_Bact_Spot	1608	459	230
Peach Healthy	-	Pec_Healthy	252	72	36
Pepper Bell Bacterial Spot	Bacteria	Pep_Bact_Spot	698	199	100
Pepper Bell Healthy	-	Pep_Healthy	1034	297	147
Potato Early Blight	Fungi	Po_E_Blight	700	200	100
Potato Healthy	-	Po_Healthy	107	30	15
Potato Late Blight	Infection	Po_L_Blight	700	200	100
Raspberry Healthy	-	Ras_Healthy	260	74	37
Soybean Healthy	-	Soy_Healthy	3563	1018	509
Squash Powdery Mildew	Fungi	Sq_Powdery	1285	367	183
Strawberry Healthy	-	Straw_Healthy	319	91	46
Strawberry Leaf Scorch	Fungi	Straw_Scorch	776	222	111
Tomato Bacterial Spot	Bacteria	Tom_Bact_Spot	1488	426	213
Tomato Early Blight	Fungi	Tom_E_Blight	700	200	100
Tomato Healthy	-	Tom_Healthy	1114	318	159
Tomato Late Blight	Infection	Tom_L_Blight	1336	382	191
Tomato Leaf Mold	Fungi	Tom_L_Mold	667	190	95
Tomato Septoria leaf Spot	Fungi	Tom_Sept	1240	354	177
Tomato Spider Mites	Mite	Tom_Sp_Mite	1174	335	167
Tomato Target Spot	Fungi	Tom_Target	984	280	140
Tomato Mosaic Virus	Virus	Tom_Mosaic	262	74	37
Tomato Yellow Leaf Curl Virus	Virus	Tom_Curl	3750	1071	536

2.4. Deep Learning Meta-Architectures

In this research, three successful DL meta-architectures were considered for the detection of plant disease. These models consist of a base network and a feature extractor. The following sub-sections provide an insight of these DL architectures to elaborate their functionality for performing an image recognition/identification task along with the overall loss function of the respective models.

2.4.1. Single Shot MultiBox Detector (SSD)

The SSD model is simple due to the elimination of the region proposal and subsequent pixel or resampling of features. This DL model includes all computations in one network, which is why it is known as a single-shot detector [31]. Experimental findings on the MS COCO, ILSVRC, and PASCAL VOC datasets revealed that the SSD achieved comparatively better precision than the other DL models such as Faster RCNN, and much faster computation time while providing a unified training and inference framework [31]. The key feature of SSD is the use of small convolution filters, such as 4x4 and 8x8; feature maps for category score; and box offset prediction for the collection of default bounding boxes. The conceptual diagram of the SSD model is presented in Figure 4. The overall loss for SSD architecture is evaluated by Equation (1) [31]:

$$L(x, c, l, g) = \frac{1}{N}[L_{conf}(x, c) + \alpha L_{loc}(x, l, g)] \quad (1)$$

where, N, L_{conf}, α, and L_{los} represent the number of matched default boxes, confidence loss, weight term, and localization loss, respectively.

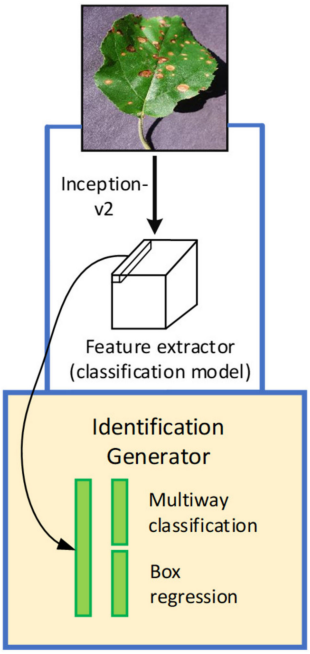


Figure 4. The basic architecture of the Single Shot MultiBox Detector (SSD) [32].

2.4.2. Faster Region-based Convolutional Neural Network (Faster-RCNN)

In Faster RCNN architecture, the object detection task performs at two different stages as compared to SSD. At the region proposal network (RPN) stage, the images are processed to generate region proposals directly through feature extractors (Inception and ResNet) instead of an external algorithm such as Edge Boxes. These features are used to forecast class-specific proposals for each intermediate convolutional layer. Then, the generated anchor boxes are used at the second step of detecting the

characteristics of the same immediate layer of an image. Figure 5 presents the basic concept of the Faster RCNN model, and the overall loss is evaluated by Equation (2) [33]:

$$L[\{p_i\},\{t_i\}] = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i, t_i^*) \quad (2)$$

where i indicates the index of anchor (responsible for obtaining bounding boxes of various sizes/ratios and used as a reference while predicting object locations); p_i, p_i^*, t_i, N_{cls}, λ, L_{cls}, L_{reg}, and N_{reg} present the output score from classification branch for anchor i, ground truth label (0/1), output prediction of the regressor layer (which consists of 4 variables (t_x, t_y, t_w, t_h)), number of anchors in mini-batch, balancing parameter, classification loss, the regressor loss (it actuates only if anchor contains an object that is ground truth (p_i^*) is 1), and number of anchors in mini-batch, respectively. Here, t_i^* is the ground truth box with a positive anchor.

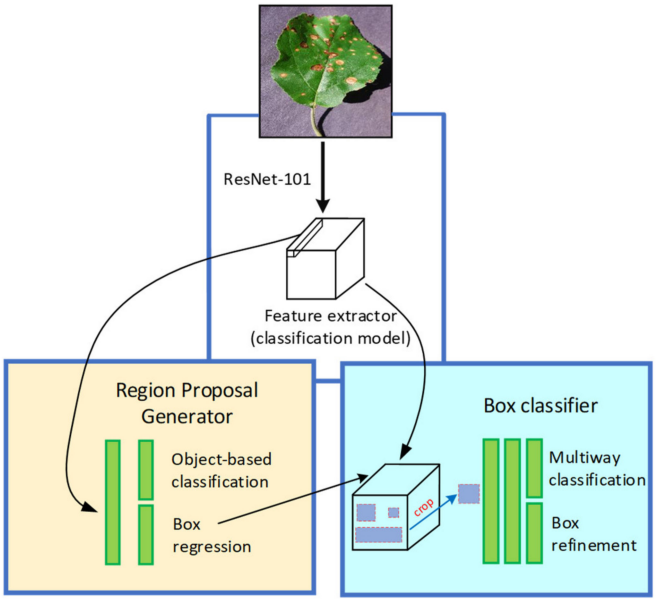


Figure 5. The basic architecture of the Faster Region-based Convolutional Neural Network (RCNN) [32].

2.4.3. Region-based Fully Convolutional Networks (RFCN)

This network is much like the Faster-RCNN, except for the removal of fully convolutional layers after the region of interest (ROI) pooling. After ROI pooling, the region proposals generate the same set of score maps for average voting. Moreover, this DL architecture has a lesser complexity level because there is no learnable layer after ROI which significantly reduces its computation time as compared to the models such as Faster RCNN. Figure 6 presents the basic concept of RFCN architecture with its corresponding proposal generator and feature extractor. The overall loss expression can be seen in Equations (3) and (4) [34]:

$$L(s, t_{x,y,w,h}) = L_{cls}(s_c^*) + \lambda[c^* > 0]L_{reg}(t, t^*) \quad (3)$$

where L_{cls} (s_c^*) is a cross-entropy loss for classification and calculates by:

$$L_{cls} = -log(s_c^*) \quad (4)$$

In Equation (3), λ is the balance weight, which is set to 1; $[c^* > 0]$ is an indicator, which is equal to 1 if the argument is true and 0 otherwise. $L_{reg}(t,t^*)$ is the bounding box regression loss and evaluated by

$smooth_{L1}$ function, c^* indicates the ground-truth label of the region of interest (ROI), and t^* presents ROI's ground truth box.

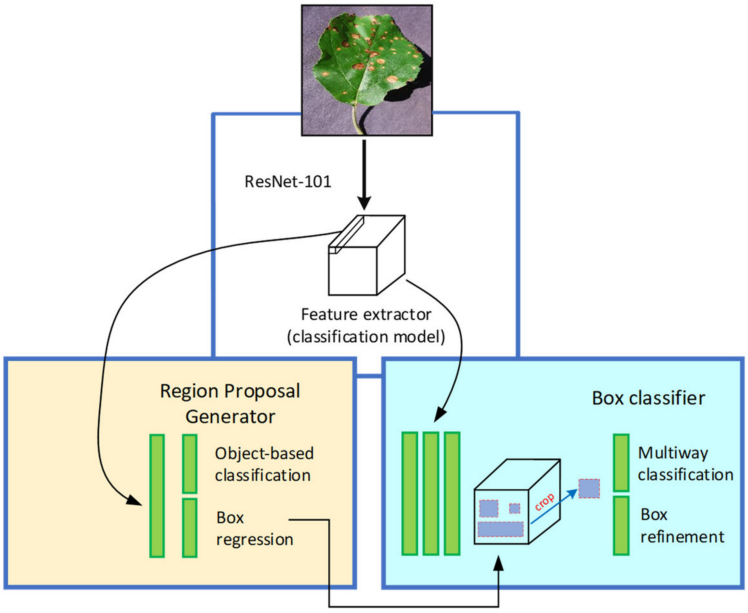


Figure 6. The basic architecture of the Region-based Fully Convolutional Network (RFCN) [32].

2.5. Deep Learning Optimizers

This article also attempted an improvement in the performance of DL meta-architectures by using various deep learning optimizers. The basic concept of these optimizing functions along with the mathematical details are presented in the following sub-sections.

2.5.1. Stochastic Gradient Descent (SGD) with Momentum

The gradient descent is the most widely used optimization algorithm for neural networks [35]. Its momentum version has faster convergence ability than the standard algorithm. The basic idea is to calculate the exponentially weighted average of the gradients and use the gradients to update the weights. To optimize the cost function, gradient descent slowly oscillates the loss towards the minimum, this slows down gradient descent and avoids by a large learning rate. However, if a larger learning rate is used, then it might end up with problems such as overshooting and diverging output. In contrast with the SGD optimizer, which used dw (calculated gradient of the weights) and db (calculated gradient of the biases) independently, the exponentially weighted averages of dw and db are taken for the momentum algorithm by the following equations (Equations (5) and (6)):

$$Vdw = \beta * Vdw + (1 - \beta) * dw \tag{5}$$

$$Vdb = \beta * Vdb + (1 - \beta) * db \tag{6}$$

where, β indicates the momentum that should be higher to smooth the update, and its default value is 0.9. Vdw and Vdb are weighted averages of optimization parameters weights and biases, respectively. After obtaining the exponentially weighted averages, weights and biases are updated by (Equations (7) and (8)):

$$W = W - lr * Vdw \tag{7}$$

$$b = b - lr * Vdb \tag{8}$$

where lr, W and b are learning rate, weight, and bias, respectively.

Plants **2020**, *9*, 1451

2.5.2. Root Mean Square Propagation (RMSProp)

The RMSProp optimization algorithm [36] limits the oscillations that generate during the loss optimization in the direction of bias, which helps to consider a larger learning rate without having an overshooting problem in training of the model. The difference between momentum and RMSProp lies in the calculation of their gradients, and weight/bias updates as shown below (Equations (9)–(12)):

$$Vdw = \beta * Vdw + (1 - \beta) * dw^2 \tag{9}$$

$$Vdb = \beta * Vdb + (1 - \beta) * db^2 \tag{10}$$

$$W = W - lr * \frac{dw}{\sqrt{vdw} + \xi} \tag{11}$$

$$b = b - lr * \frac{db}{\sqrt{vdb} + \xi} \tag{12}$$

The Vdw of the RMSProp optimizer could be relatively small (even 0); therefore, epsilon (ξ) adds in the denominator for numerical stability. When Vdw is relatively small, it increases weights *(W)*, and then the updates in the direction of weights become fast. However, Vdb is relatively large, which decreases bias *(b)* to slow down the updates in its direction.

2.5.3. Adaptive Moment Estimation (Adam)

The idea behind the Adam optimization algorithm is taking momentum and RMSProp and putting them together [37]. It means that the Adam optimizer retains an exponentially decaying average gradient of the previous gradients as well as previously squared gradients. First, the initialization of Vdw, Vdb, Sdw (element-wise squaring of Vdw), and Sdb (element-wise squaring of Vdb) is set to zero. Then, for a certain number of iterations, this algorithm computes the dw and db using current mini-batch and performs exponentially weighted average by using Equations (5), (6), (9), and (10):

Then, the calculations for the corrected Vdw, Vdb, Sdw, and Sdb are performed for bias correction by the following equations (Equations (13)–(16)):

$$Vdw^{corrected} = Vdw/(1 - \beta_1 t) \tag{13}$$

$$Vdb^{corrected} = Vdb/(1 - \beta_1 t) \tag{14}$$

$$Sdw^{corrected} = Sdw/(1 - \beta_2 t) \tag{15}$$

$$Sdb^{corrected} = Sdb/(1 - \beta_2 t) \tag{16}$$

where, β_1 and β_2 indicate exponential decay rate for the first moment and second moment, respectively.

Weight (W) and bias (b) are updated by (Equations (17) and (18)):

$$W = W - lr\frac{Vdw^{corrected}}{\sqrt{Sdw^{corrected}} + \xi} \tag{17}$$

$$b = b - lr\frac{Vdb^{corrected}}{\sqrt{Sdb^{corrected}} + \xi} \tag{18}$$

2.6. Experimental Setup

The experiments are based on three popular DL meta-architectures: SSD, Faster-RCNN, and R-FCN, which were previously trained on 1.5 million images (80 categories) of the Common Objects in Context (COCO) dataset. The transfer learning technique was used to obtain better detection results. First, the trained layers were frozen to reuse some fundamental features such as corners, borders, and edges; then, a few new and workable layers were added that learned the specific features of the new

dataset (PlantVillage). The backbone architectures named Inception-v2 [38], Inception ResNet-v2 [39], and versions of ResNet including ResNet-50 and ResNet-101 [40] were used with the base networks to classify and localize the plant disease. Table 2 presents the base networks with feature extraction methods along with their performance measured in mAP on the COCO dataset.

Table 2. List of base DL models with feature extraction methods along with mean average precision (mAP) on the Common Objects in Context (COCO) dataset.

Base Networks	Feature Extraction Methods	mAP (%) for COCO Dataset
SSD	Inception v2	24
Faster-RCNN	Inception v2	28
	ResNet-50	30
	ResNet-101	32
	Inception-ResNet v2	37
R-FCN	ResNet-101	30

Table 3 presents the hyperparameters: α (learning rate), β_1 known as the first moment has a default value of 0.9, β_2 known as the second moment has a default value of 0.999, and epsilon (ξ)= 10^{-8}. However, the default value of ξ may not be sufficient in general for some machine learning problems. The best learning rate was selected from three sets (10^{-4}, 10^{-5}, 10^{-6}) to determine the efficacy of the initializations. These learning rates were considered for different steps of iterations while training [31,33,37]. The tuning of the hyperparameters of all the DL optimizers was performed using the random search technique [15,41]. All the DL models were trained using Graphics Processing Units (NVIDIA GTX 1650 and 1050) for high-performance acceleration.

Table 3. List of hyperparameters of the respective DL optimizers.

DL Optimizers	*lr*	Momentum	β/β_1	β_2	ξ
SGD with Momentum (default)	0.01	0.9	-	-	-
SGD with Momentum (modified)	0.0003	0.9	-	-	-
Adam (default)	0.001	-	0.9	0.999	1×10^{-08}
Adam (modified)	0.0002	-	0.9	0.9997	1×10^{-03}
RMSProp (default)	0.001	0.0	0.9	-	1×10^{-08}
RMSProp (modified)	0.0004	0.0	0.95	-	1×10^{-02}

2.7. Performance Metric

The performance of the DL-based plant disease detectors was evaluated by using mean average precision (mAP). This performance metric is commonly used with the DL meta-architectures (SSD, Faster-RCNN) to detect artifacts such as COCO [29] and PASCAL's VOC challenge [42]. Any algorithm providing the predicted bounding boxes as an output can be assessed with intersection of union (IoU), Average Precision (AP), and mAP [42]. The x, y coordinates require (*Xmin, Xmax, Ymin,* and *Ymax*) to track the efficiency of the DL architectures. In Figure 7, an image belongs to a strawberry leaf class is presented, where the DL model provided an output in the form of a predicted bounding box with scorch disease on the image. Two bounding boxes can be seen in Figure 7: one shows the exact location of the healthy/defected part in an image, named as the ground-truth bounding box. Another is an actual predicted bounding box that is drawn by the trained DL model.

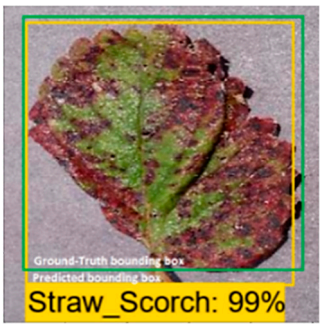


Figure 7. Visual example of ground truth bounding box (green) versus predicted bounding box (yellow) for intersection of the union (IoU) metric.

For evaluating the mAP, it should be noted that the precision measures how accurate the predictions are—that is, the percentage of the correct predictions—and recall measures how well all positive outcomes are found. The average precision (AP) was assessed with an 11-point interpolated average precision method. The precision and recall were computed for each class. The AP is the average precision across all unique recall levels. Before measuring AP, we first interplay the precision at multiple recall levels. At a certain recall level r, the interpolated precision (p_{int}) is specified as the highest precision for a recall level $r' \geq r$ [24,42] (Equations (19) and (20)).

$$p_{\text{int}}(r) = \max p(r') \tag{19}$$

where $p(r')$ is the measured precision at the max recall rr'.

The AP is then described as the mean precision at the eleven recall rates equally spaced [0, 0.1, 0.2, ..., 1].

$$AP = \frac{1}{11} \sum_{r \in \{0, 0.1, 0.2, \ldots 1\}} p_{\text{int}}(r) \tag{20}$$

The mean average accuracy (mAP) is the approximate average value of all individual APs. To evaluate the mAP, the AP of each class was first calculated (as described above). Then, the mAP was found by Equation (21):

$$mAP = \frac{\sum_{i=1}^{n} AP_i}{n} \tag{21}$$

where n = 38 (number of classes).

3. Results and Discussion

The goal of this research is not only the identification of the presence of diseased and healthy leaves but also to locate a confidence score indicating the likelihood that there is a correct (true positive) class in a bounding box. The score was considered between 0 and 1 (or 0–100%), indicating how much precisely the type of plant disease was recognized. It was empirically observed that all the DL meta-architectures required 126 epochs (200,000 iterations) to converge their training. The loss plots of each DL architecture with its detection results are presented in this section. Moreover, the improvement in the performance of the best-suited architectures is also presented. The mAP attained by each DL meta-architecture with its corresponding optimizer is shown in Table 4.

Plants **2020**, *9*, 1451

Table 4. Summary of plant disease identification results indicating the Average Precision (AP) for each leaf class and the overall mAP for each DL meta-architecture. The hyphen (-) denotes the failed detection in the respective classes.

Annotated Class Labels	DL Meta-Architectures with Feature Extractors and Optimizers									
	R-FCN ResNet-101	Faster-RCNN						SSD Inception-v2		
		ResNet-50	Inception ResNet-v2	Inception-v2	ResNet-101					
	SGD with Momentum	SGD with Momentum	SGD with Momentum	SGD with Momentum	SGD with Momentum	RMSProp	Adam	SGD with Momentum	RMSProp	Adam
A_Scab	90.61	-	90.04	31.22	61.37	99.83	22.93	27.24	36.16	27.27
A_Blk_Rot	47.36	89.69	-	24.6	41.82	57.41	43.82	45.42	45.45	54.55
A_C_Rust	90.24	65.33	-	7.27	34.98	91.64	35.48	45.45	45.45	54.55
A_Healthy	90.85	99.97	2.19	75.91	43.42	6.71	18.13	90.91	100	90.91
B_Healthy	99.97	-	100	85.58	90	22.77	88.8	100	90.91	100
Ch_Healthy	79.22	-	-	87.32	75.69	99.84	63.66	90.86	90.91	90.91
Ch_Mildew	40.46	-	-	99.62	99.55	95.39	98.87	36.36	45.45	45.45
Corn_Rust	0.76	-	-	99.85	99.89	100	99.63	90.91	99.89	99.92
Corn_Healthy	-	-	-	100	99.96	76.13	90.87	90.91	90.91	90.91
Corn_Blight	42.57	-	-	51.01	66.34	93.53	96.09	45.45	54.55	54.29
Corn_Spot	0.31	-	-	1.15	20.56	31.83	10.95	0.73	5.62	33.13
G_Blk_Rot	7	-	-	1.16	53.9	53.87	0.05	71.26	72.73	72.73
G_Blk_Measles	0.09	-	-	100	100	100	99.61	100	90.91	99.87
Grp_Healthy	-	-	-	99.59	99.3	98.86	90.55	100	90.91	100
Grp_Blight	0.21	-	-	72.73	99.93	94.29	81.06	72.73	72.73	72.73
O_HLBing	12.74	-	-	99.99	90.91	98.14	90.91	90.91	90.91	90.91
Pec_Bact_Spot	0.28	-	-	14.23	40.23	74.16	-	9.09	18.14	27.23
Pec_Healthy	-	-	-	8.21	34.11	42.93	5.34	90.91	99.28	100
Pep_Bact_Spot	6.8	-	-	1.86	2.58	35.61	-	9.09	18.11	36.27
Pep_Healthy	50.95	-	-	6.45	2.11	19.08	2.33	90.91	90.8	90.91
Po_E_Blight	59.95	-	-	-	2.51	32	-	9.09	9.09	26.84
Po_Healthy	-	-	-	-	-	0.18	-	90.91	90.61	90.91
Po_L_Blight	94.77	-	-	-	-	1.79	-	29.16	43.81	44.92
Ras_Healthy	0.23	-	-	0.33	1.6	9.53	1.14	90.91	100	100
Soy_Healthy	88.11	-	-	26.03	59.43	9.6	3.35	90.91	90.13	90.91
Sq_Powdery	99.46	-	-	52.68	99.4	5.65	54.61	81.82	90.91	81.82

Table 4. *Cont.*

Annotated Class Labels	DL Meta-Architectures with Feature Extractors and Optimizers									
	R-FCN ResNet-101	Faster-RCNN						SSD Inception-v2		
		ResNet-50	Inception ResNet-v2	Inception-v2	ResNet-101					
	SGD with Momentum	SGD with Momentum	SGD with Momentum	SGD with Momentum	SGD with Momentum	RMSProp	Adam	SGD with Momentum	RMSProp	Adam
Straw_Healthy	99.3	-	-	53.33	18.07	1.1	62.53	100	90.91	100
Straw_Scorch	100	98.34	-	72.62	70.47	86.45	6.94	72.66	72.7	72.69
Tom_Bact_Spot	98.85	-	99.93	0.29	2.3	5.57	-	18.03	26.67	36.28
Tom_E_Blight	-	-	-	7.04	39.38	64.2	11.59	27.12	36.36	54.45
Tom_Healthy	0.2	-	-	100	87.13	49.65	100	100	100	100
Tom_L_Blight	-	-	-	99.96	99.96	95.92	90.36	81.79	81.72	90.77
Tom_L_Mold	3.87	-	-	96.21	98.7	99.55	82.21	45.41	63.6	63.64
Tom_Sept	-	-	-	99.56	95.24	100	99.83	90.86	90.88	90.91
Tom_Sp_Mite	85.12	-	-	98.52	98.06	98.73	99.9	90.88	90.88	90.91
Tom_Target	1.12	-	-	61.56	99.98	83.77	96.41	35.81	35.24	45.4
Tom_Mosaic	98.15	-	-	9.8	85.98	22.17	-	72.73	54.55	63.64
Tom_Curl	85.48	-	-	99.68	99.98	99.86	100	100	100	100
mAP (%)	41.45	9.30	7.69	51.19	60.92	59.41	48.63	66.51	68.89	73.07

3.1. Performance of Deep Learning Meta-Architectures

During the first phase of the proposed approach, all the DL architectures were trained with the momentum optimizer due to its fast convergence ability [43]. The SSD model outperformed Faster RCNN and RFCN models. Further explanations of the results of each model are provided as follows:

3.1.1. SSD Architecture

An input image of 300×300 size was considered for all the experiments. The SSD architecture was trained with the feature extraction method called Inception-v2 with different learning rates. The model was trained by using SGD with the momentum optimizer using the learning rate as 3×10^{-4}, $3 \times 10^{-5,}$ and 3×10^{-6} for 90k, 30k, and 80k iterations, respectively. This model took approximately 4.25 days to complete its training. The training loss curve of the SSD model is shown in Figure 8. At the end of the training, the loss curve indicated its fluctuation between 0.64% and 3.73%. After the training of the SSD model, the images from the testing dataset were used to classify and localize the defected spots of plant disease. Figure 9 shows the detection results obtained by the SSD along with their confidence score. Table 4 indicates the average precision of each leaf category and the mAP of 38 leaf disease categories. The mAP obtained by this state-of-the-art DL architecture was 66.51%, which is the highest among all the other models. It is noticed that the results of six plant classes such as blueberry healthy, grape healthy, grape black measles, strawberry healthy, tomato healthy, and tomato curl virus were quite promising, due to their 100% average precision. For around 12 leaf categories, the average precision was more than 90%. Around 14 disease classes achieved low precision (less than 50%). It is also noticed that the precision of corn gray leaf spot was the lowest among all the other classes of plant disease, which was addressed in the next step of the proposed method.

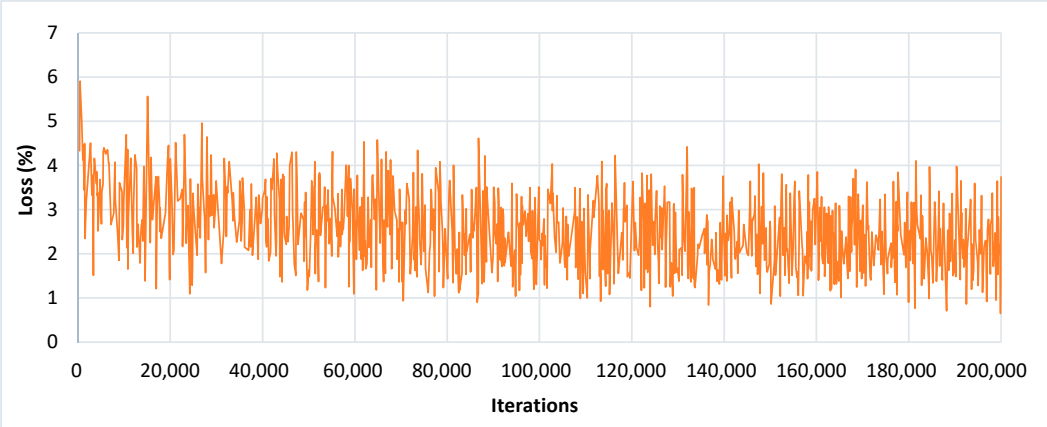


Figure 8. Training loss curve of SSD with Inception-v2 model.

Plants **2020**, *9*, 1451

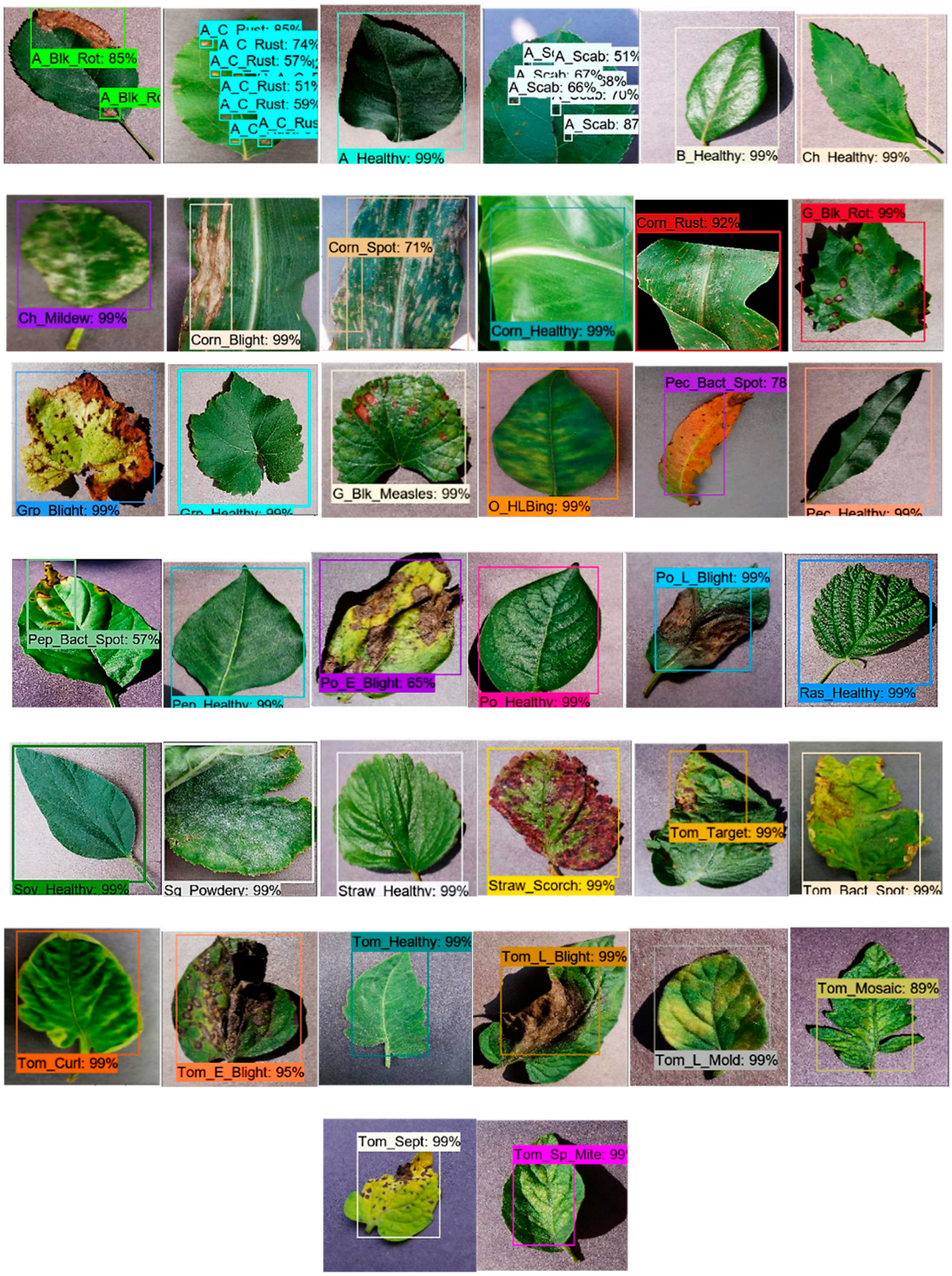


Figure 9. The detection results by the SSD with the Inception-v2 model having a threshold score of 0.5. Each predicted box is associated with a category label having a confidence score between 0 and 100%.

3.1.2. Faster-RCNN Architecture

The Faster-RCNN model was trained with the feature extractors including ResNet-50, ResNet-101, Inception-v2, and Inception ResNet-v2. All the feature extraction methods were trained with momentum optimizer using learning rate 3×10^{-4}, $3 \times 10^{-5,}$ and 3×10^{-6} for 90k, 30k, and 80k iterations, respectively. Initially, the Faster RCNN model was trained with the feature extraction methods such as ResNet-50 and Inception Resnet-v2, but they failed to detect and localize most of the classes (as shown in Figure 10,

Figure 11, and Table 4) and resulted in the lowest mAP among all the DL architectures. This was due to the presence of some challenging disease categories (potato early blight, potato late blight, tomato early blight, and tomato late blight, etc.), since specific features such as leaf shape, disease spots, and colour of disease spot were quite similar.

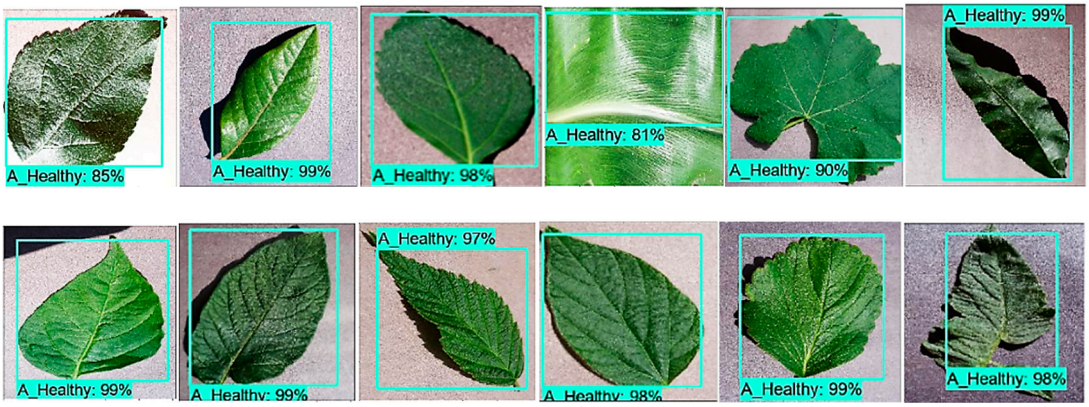


Figure 10. False positive detection results of healthy classes by Faster-RCNN with ResNet-50 architecture. The first image is from the apple healthy category, which is the only class that the model detects correctly. Other images were originally from blueberry, cherry, corn, grape, peach, pepper bell, potato, raspberry, soybean, strawberry, and tomato healthy classes, but the model identified them as Apple healthy class.

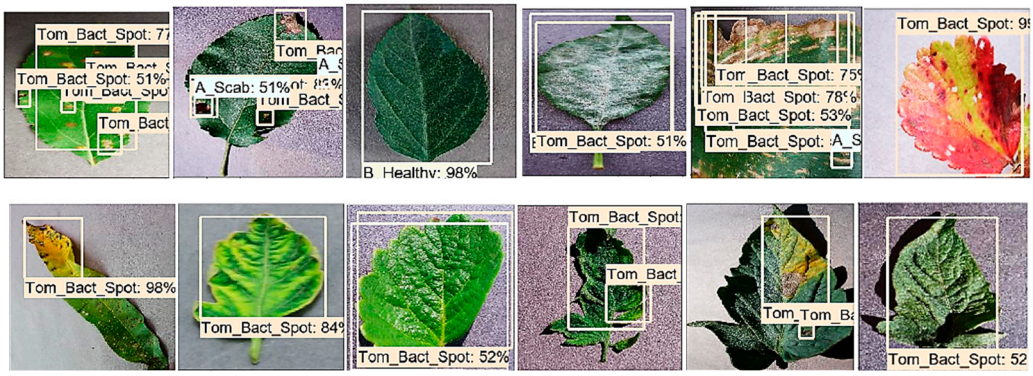


Figure 11. Some of the examples of false positives detection results using the Faster-RCNN with Inception ResNet-v2 architecture. It identified most of the leaf categories as the tomato bacterial spot.

To obtain a noticeable improvement in the identification results by the Faster-RCNN model, two other classification models/feature extractors (Inception-v2 and ResNet-101) were also considered. Due to different feature extractors, the training time of the Faster RCNN model was varied. For example, ResNet-50 required the lowest training time of around 34 h, whereas Inception-v2 was the slowest among the other classification models as it took approximately 48 h to complete its training. However, ResNet-101 and Inception ResNet-v2 needed 37.5 and 44.20 h, respectively. It was observed that the baseline Faster-RCNN performed well when combined with the feature extraction method ResNet-101. It achieved 60.92% mAP@0.5, which is 9.73% higher than with the Inception-v2 (51.19%) with the same training settings. The training loss obtained by the Faster RCNN with the ResNet-101 model is presented in Figure 12, and its lower percentage error proved the effective learning of the specific features of plant disease after 200k iterations. From Table 4, it can be observed that the Faster RCNN with ResNet-101 architecture identified Grape Black Measles disease with 100% precision. Moreover, it attained more than 90% precision for almost 14 leaf classes. However, 13 classes achieved a precision of less than 50%. A further five classes were poorly detected and resulted in less than 10% precision. It is also noticed that two classes (potato healthy and potato late blight) failed to detect by Faster-RCNN with the ResNet-101 model.

Plants **2020**, *9*, 1451

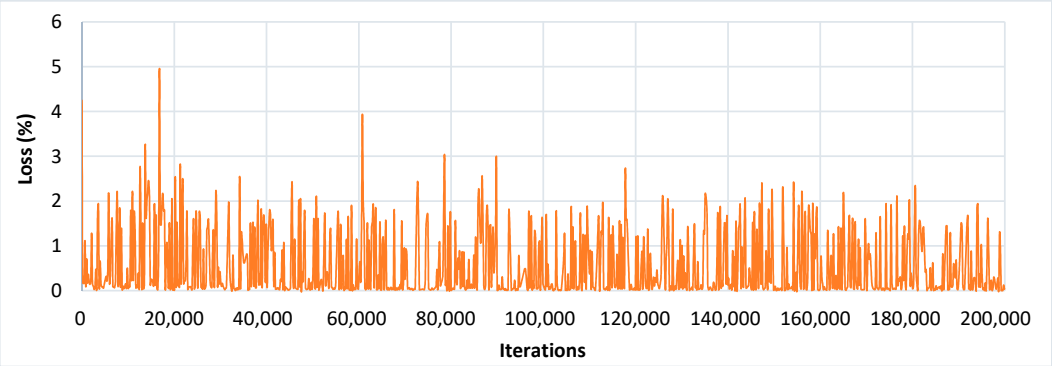


Figure 12. Training loss curve of Faster-RCNN with ResNet-101 model.

3.1.3. R-FCN Architecture

Using the ResNet-101 as the feature extractor, the RFCN model achieved good detection results of 83.6% mAP on the PASCAL VOC dataset. Therefore, the same feature extractor with the baseline model (RFCN) was also considered in this research. In this DL method, all learnable weight layers are convolutional, which computes the feature map on the entire image. The R-FCN model was fine-tuned and trained with the momentum optimizer using the learning rate 3×10^{-4} for 90k steps, and then continued training for the next 30k steps with 3×10^{-5}, and finally, 80k steps with 3×10^{-6}. This model completed 200k iterations in 33.7 h. The training loss obtained by the RFCN model is presented in Figure 13, and its percentage loss oscillated from 0.03% to 1.28% after 200k iterations.

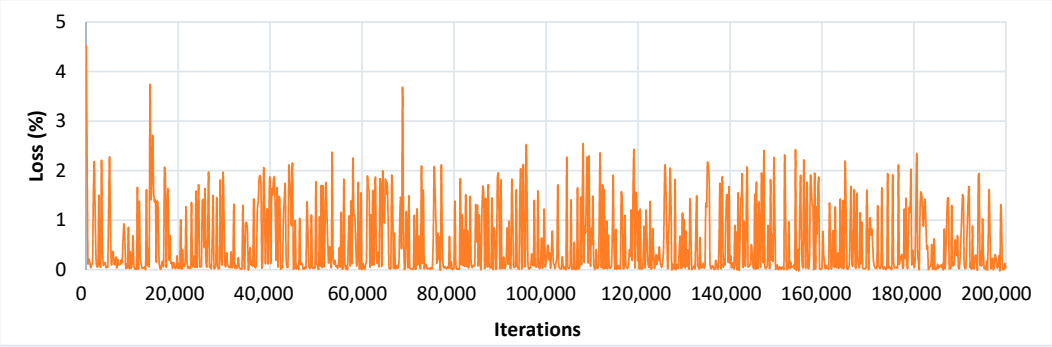


Figure 13. Training loss curve of the R-FCN model.

This model identified a lesser number of classes than the SSD and Faster RCNN (with ResNet-101) models; the failed classes were: corn healthy, grape healthy, peach healthy, potato healthy, tomato Septoria, tomato early blight, and tomato late blight (as shown in Table 4). The strawberry scorch class achieved the perfect average precision of 100%. It is also observed that 9 plant classes provided good detection results and achieved considerable precision (more than 90%). Around 11 other classes showed an average precision of less than 10%. Few examples of the false/confused detection results by the RFCN model are shown in Figure 14.

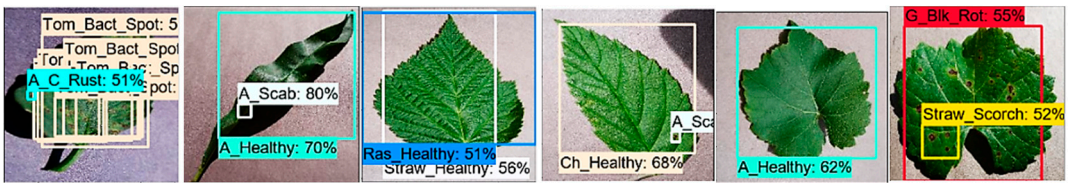


Figure 14. Some examples of false-positive detection results by the RFCN model.

Plants **2020**, *9*, 1451

3.2. Overall Remarks for SSD, Faster RCNN, and RFCN Architectures

From Table 4, important observations regarding the performance of the DL meta-architectures are presented as follows:

- The SSD model achieved the highest mAP among all the DL meta-architectures. This is due to the structural behaviour of the SSD model which provides a fixed-size predictive box set and scores at each feature-layer position of a kernel. The convolutional layers are added to the last of the base network which predicts multiple scales [31]. The projected performance value boxes in each feature map location compared to the default position boxes are determined using an intermediate connected layer in these positions instead of a fully convolution layer.
- Another significant distinction of the SSD model is that the information in ground-level truth boxes allocates to different outputs within the defined collection of detector outputs during SSD training compared to other regional networks. The structure of the network decides which ground box should be matched with its corresponding default box during the training stage, known as matching strategy in SSD. Thereby, the use of several convolutional bounding box outputs connected to features maps at the top of the network made this model successful as compared to other region-based methods.
- The base network SSD combined with the" Inception" model performed better than the Faster-RCNN combined with the same feature extraction method. Moreover, Table 4 shows that the base network Faster-RCNN with feature extractor ResNet-101 showed relatively higher mAP than with the Inception model.
- The RFCN model achieved lower mAP than the SSD and Faster RCNN (with ResNet-101) architectures.
- More interestingly, the SSD architecture was able to detect few of those classes that were completely undetected by the Faster RCNN and RFCN models (as shown in Table 4).
- Following the proposed methodology presented in Section 2, the SSD with Inception-v2 and Faster RCNN with ResNet-101 models achieved the highest mAP among all the other DL meta-architectures. Therefore, they were selected for the next stage of this research.

3.3. Performance Improvement by DL Optimization Algorithms

After obtaining two best-suited DL meta-architectures, better optimization of the weight parameters was attempted by Adam and RMSProp optimization algorithms. Their learning rate is presented in Table 3. Table 4 presents the change in mAP for both the selected models. Some concluding remarks are provided as follows:

- The Faster-RCNN with the ResNet-101 model trained by Adam and RMSProp optimizers failed to improve its overall mAP as compared to the SGD (with momentum) optimizer.
- On the other hand, the SSD model achieved 66.51% mAP when it was trained by the momentum optimizer. Then, its mAP was increased by about 2.38% with the RMSProp optimizer. Further improvement of 3.39% in the mAP was observed when the weights of the SSD model were optimized by the Adam optimization algorithm.
- It is also noticed that when the SSD model was trained by Adam optimizer, the average precision of several leaf categories significantly improved, due to which the highest mAP of 73.07% was attained. The AP of classes such as Apple black rot, Apple cedar rust, Tomato early blight, and disease was increased to more than 50%. The AP of few other classes also improved (but still less than 50%) including Tomato target spot, Tomato bacterial spot, Potato late blight, Potato early blight, Pepper bacterial spot, and Peach bacterial spot. The AP of corn gray leaf spot class also improved, which was previously unsuccessful in providing a noticeable AP when the dataset was trained with the SGD with momentum and RMSProp optimizers. However, the further improvement in AP should be considered in future research.

Plants **2020**, *9*, 1451

- Figure 15 presents the change in AP for each class of the PlantVillage dataset when they were trained by the SSD model with all the three DL optimizers.
- A summary of the mAP achieved by DL meta-architectures trained with different optimization algorithms is presented in Figure 16.

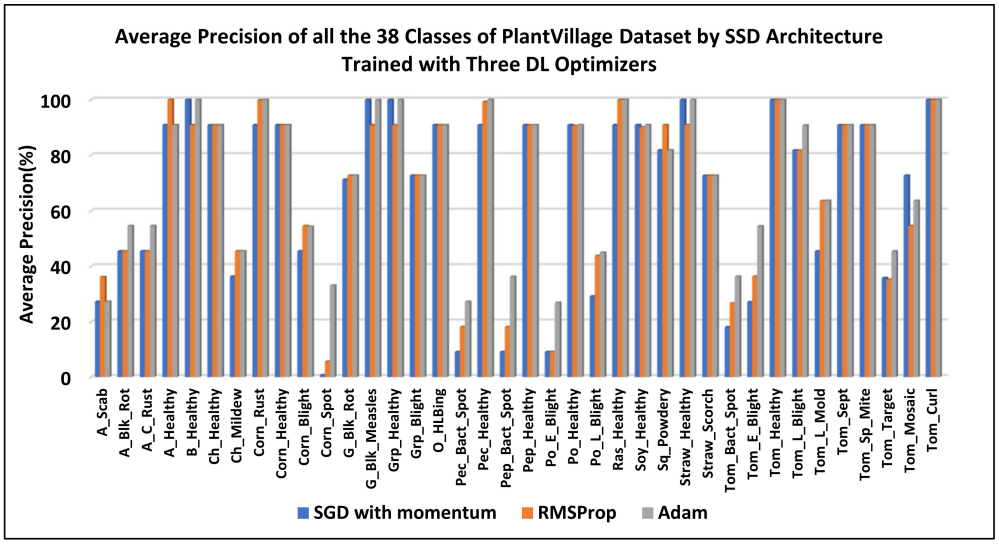


Figure 15. Performance plot of the SSD model trained with three DL optimizers.

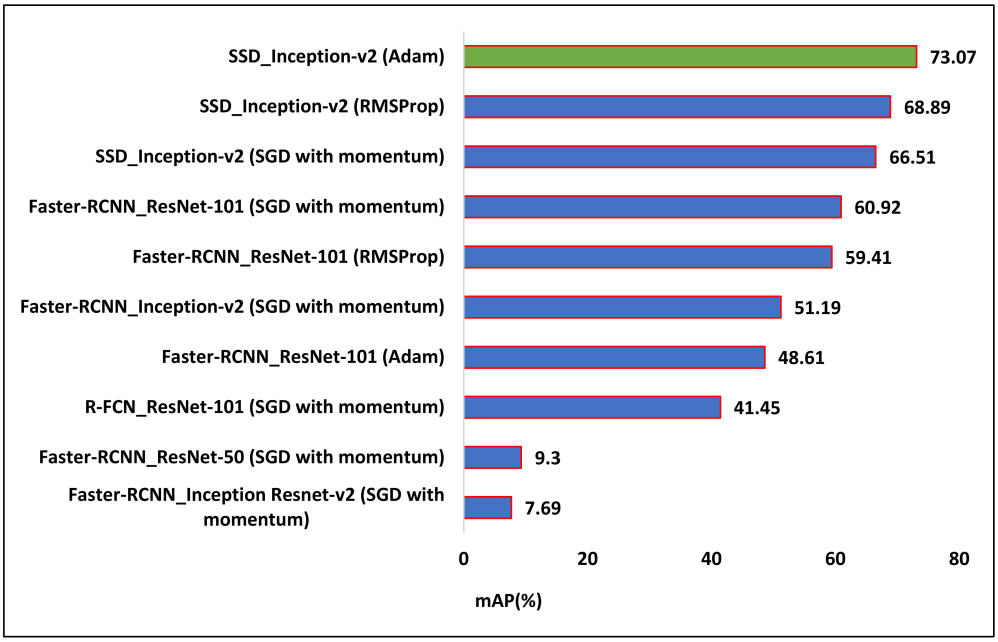


Figure 16. Summary of mean average precision (%) achieved by the DL meta-architectures with their respective optimizers (the best combination of DL meta-architecture and optimizer is shown by green bar).

4. Conclusions and Future Work

The main goal of this research was to perform the complex task of plant disease localization and classification in a single framework. In this regard, state-of-the-art deep learning meta-architectures including SSD, Faster RCNN, and RFCN models were trained and tested on 38 different classes of healthy/defected plant leaves. Moreover, an improvement in their performance was also attempted by better optimization of weight parameters through Adam and RMSProp optimizers. The SSD

model trained with the feature extractor Inception-v2 attained the highest mean average precision as compared to the other DL meta-architectures. It achieved the best identification results by training through an Adam optimizer and attained 73.07% of mAP. All the healthy/diseased leaf classes were identified, which proves the novelty of the proposed approach. Practically, the successful detection of plant disease by DL technique would be useful to reduce the undesirable application of fungicide spray.

Few future recommendations for the research community are presented as follows:

- The trained and tested DL models' pipeline, checkpoints, and weights can be reused as a transfer learning approach for upcoming researches related to plant disease detection.
- Various factors affecting the performance of best-suited DL architecture should be investigated such as data augmentation techniques, batch size, aspect ratios, etc.
- Although, all the classes of the PlantVillage dataset were identified by the proposed methodology; still, few of them achieved a lower average precision. Therefore, few modifications in DL networks can also be proposed in the future to further improve the mean average precision.
- This research could also be beneficial for several robotic systems to identify/classify healthy and unhealthy crops in real-time that would contribute to agricultural automation.

Author Contributions: Conceptualization, methodology, investigation, visualization, M.H.S., S.K., and K.M.A.; writing—original draft preparation, M.H.S. and S.K.; writing—review and editing, M.H.S., S.K., and K.M.A; supervision, project administration, J.P. and K.M.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of Business, Innovation and Employment (MBIE), New Zealand, Science for Technological Innovation (SfTI) National Science Challenge.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sankaran, S.; Mishra, A.; Ehsani, R.; Davis, C. A review of advanced techniques for detecting plant diseases. *Comput. Electron. Agric.* **2010**, *72*, 1–13. [CrossRef]
2. Duro, D.C.; Franklin, S.E.; Dubé, M.G. A comparison of pixel-based and object-based image analysis with selected machine learning algorithms for the classification of agricultural landscapes using SPOT-5 HRG imagery. *Remote Sens. Environ.* **2012**, *118*, 259–272. [CrossRef]
3. Yamamoto, K.; Guo, W.; Yoshioka, Y.; Ninomiya, S. On plant detection of intact tomato fruits using image analysis and machine learning methods. *Sensors* **2014**, *14*, 12191–12206. [CrossRef] [PubMed]
4. Esteva, A.; Robicquet, A.; Ramsundar, B.; Kuleshov, V.; DePristo, M.; Chou, K.; Cui, C.; Corrado, G.; Thrun, S.; Dean, J. A guide to deep learning in healthcare. *Nat. Med.* **2019**, *25*, 24–29. [CrossRef] [PubMed]
5. Kocić, J.; Jovičić, N.; Drndarević, V. An end-to-end deep neural network for autonomous driving designed for embedded automotive platforms. *Sensors* **2019**, *19*, 2064. [CrossRef]
6. Saleem, M.H.; Potgieter, J.; Arif, K.M. Plant disease detection and classification by deep learning. *Plants* **2019**, *8*, 468. [CrossRef]
7. Adhikari, S.P.; Yang, H.; Kim, H. Learning semantic graphics using convolutional encoder-decoder network for autonomous weeding in paddy field. *Front. Plant Sci.* **2019**, *10*, 1404. [CrossRef]
8. Olsen, A.; Konovalov, D.A.; Philippa, B.; Ridd, P.; Wood, J.C.; Johns, J.; Banks, W.; Girgenti, B.; Kenny, O.; Whinney, J. DeepWeeds: A multiclass weed species image dataset for deep learning. *Sci. Rep.* **2019**, *9*, 1–12. [CrossRef]
9. Marani, R.; Milella, A.; Petitti, A.; Reina, G. Deep neural networks for grape bunch segmentation in natural images from a consumer-grade camera. *Precis. Agric.* **2020**, 1–27. [CrossRef]
10. Wan, S.; Goudos, S. Faster R-CNN for multi-class fruit detection using a robotic vision system. *Comput. Netw.* **2020**, *168*, 107036. [CrossRef]
11. Ampatzidis, Y.; Partel, V. UAV-based high throughput phenotyping in citrus utilizing multispectral imaging and artificial intelligence. *Remote Sens.* **2019**, *11*, 410. [CrossRef]

12. Fuentes-Pacheco, J.; Torres-Olivares, J.; Roman-Rangel, E.; Cervantes, S.; Juarez-Lopez, P.; Hermosillo-Valadez, J.; Rendón-Mancha, J.M. Fig Plant Segmentation from Aerial Images Using a Deep Convolutional Encoder-Decoder Network. *Remote Sens.* **2019**, *11*, 1157. [CrossRef]
13. Quiroz, I.A.; Alférez, G.H. Image recognition of Legacy blueberries in a Chilean smart farm through deep learning. *Comput. Electron. Agric.* **2020**, *168*, 105044. [CrossRef]
14. Wu, C.; Zeng, R.; Pan, J.; Wang, C.C.; Liu, Y.-J. Plant phenotyping by deep-learning-based planner for multi-robots. *IEEE Robot. Autom. Lett.* **2019**, *4*, 3113–3120. [CrossRef]
15. Saleem, M.H.; Potgieter, J.; Arif, K.M. Plant Disease Classification: A Comparative Evaluation of Convolutional Neural Networks and Deep Learning Optimizers. *Plants* **2020**, *9*, 1319. [CrossRef]
16. Chen, J.; Liu, Q.; Gao, L. Visual Tea Leaf Disease Recognition Using a Convolutional Neural Network Model. *Symmetry* **2019**, *11*, 343. [CrossRef]
17. Kamal, K.; Yin, Z.; Wu, M.; Wu, Z. Depthwise separable convolution architectures for plant disease classification. *Comput. Electron. Agric.* **2019**, *165*, 104948.
18. Karthik, R.; Hariharan, M.; Anand, S.; Mathikshara, P.; Johnson, A.; Menaka, R. Attention embedded residual CNN for disease detection in tomato leaves. *Appl. Soft Comput.* **2020**, *86*, 105933.
19. Geetharamani, G.; Pandian, A. Identification of plant leaf diseases using a nine-layer deep convolutional neural network. *Comput. Electr. Eng.* **2019**, *76*, 323–338.
20. Vaishnnave, M.; Devi, K.S.; Ganeshkumar, P. Automatic method for classification of groundnut diseases using deep convolutional neural network. *Soft Comput.* **2020**, *24*, 16347–16360. [CrossRef]
21. Mohanty, S.P.; Hughes, D.P.; Salathé, M. Using deep learning for image-based plant disease detection. *Front. Plant Sci.* **2016**, *7*, 1419. [CrossRef]
22. Too, E.C.; Yujian, L.; Njuki, S.; Yingchun, L. A comparative study of fine-tuning deep learning models for plant disease identification. *Comput. Electron. Agric.* **2019**, *161*, 272–279. [CrossRef]
23. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the 2014 IEEE conference on computer vision and pattern recognition (CVPR), Columbus, OH, USA, 24–27 June 2014; pp. 580–587.
24. Fuentes, A.; Yoon, S.; Kim, S.C.; Park, D.S. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors* **2017**, *17*, 2022. [CrossRef]
25. Gutierrez, A.; Ansuategi, A.; Susperregi, L.; Tubío, C.; Rankić, I.; Lenža, L. A Benchmarking of Learning Strategies for Pest Detection and Identification on Tomato Plants for Autonomous Scouting Robots Using Internal Databases. *J. Sens.* **2019**, *2019*. [CrossRef]
26. Ramcharan, A.; McCloskey, P.; Baranowski, K.; Mbilinyi, N.; Mrisho, L.; Ndalahwa, M.; Legg, J.; Hughes, D.P. A mobile-based deep learning model for cassava disease diagnosis. *Front. Plant Sci.* **2019**, *10*, 272. [CrossRef] [PubMed]
27. Ji, M.; Zhang, K.; Wu, Q.; Deng, Z. Multi-label learning for crop leaf diseases recognition and severity estimation based on convolutional neural networks. *Soft Comput.* **2020**, *24*, 15327–15340. [CrossRef]
28. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in neural information processing systems (NIPS 2012), Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
29. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the European conference on computer vision (ECCV), Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
30. Hughes, D.; Salathé, M. An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv* **2015**, arXiv:1511.08060.
31. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European conference on computer vision (ECCV), Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
32. Huang, J.; Rathod, V.; Sun, C.; Zhu, M.; Korattikara, A.; Fathi, A.; Fischer, I.; Wojna, Z.; Song, Y.; Guadarrama, S. Speed/accuracy trade-offs for modern convolutional object detectors. In Proceedings of the 2017 IEEE conference on computer vision and pattern recognition (CVPR), Hawaii Convention Center, Honolulu, Hawaii, 21–26 July 2017; pp. 7310–7311.

33. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in neural information processing systems (NIPS), Montreal Convention Center, Montreal, QC, Canada, 7–10 December 2015; pp. 91–99.
34. Dai, J.; Li, Y.; He, K.; Sun, J. R-fcn: Object detection via region-based fully convolutional networks. In Proceedings of the Advances in neural information processing systems (NIPS), International Barcelona Convention Center, Barcelona, Spain, 5–10 December 2016; pp. 379–387.
35. Ruder, S. An overview of gradient descent optimization algorithms. *arXiv* **2016**, arXiv:1609.04747.
36. Hinton, G.; Srivastava, N.; Swersky, K. Neural Networks for Machine Learning. Available online: http://www.cs.toronto.edu/~{}hinton/coursera/lecture6/lec6.pdf (accessed on 7 October 2020).
37. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
38. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 2818–2826.
39. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A. Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv* **2016**, arXiv:1602.07261.
40. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
41. Bergstra, J.; Bengio, Y. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* **2012**, *13*, 281–305.
42. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
43. Brahimi, M.; Arsenovic, M.; Laraba, S.; Sladojevic, S.; Boukhalfa, K.; Moussaoui, A. Deep learning for plant diseases: Detection and saliency map visualisation. In *Human and Machine Learning*; Springer: Berlin, Germany, 2018; pp. 93–117.

Chapter 5 Weed Identification by Single-Stage and Two-Stage Neural Networks: A Study on the Impact of Image Resizers and Weights Optimization Algorithms

This chapter contains content from the following article.

M. H. Saleem, K. K. Velayudhan, J. Potgieter, and K. M. Arif, "Weed identification by single-stage and two-stage neural networks: A study on the impact of image resizers and weights optimization algorithms," *Frontiers in Plant Science*, vol. 13, 2022. https://doi.org/10.3389/fpls.2022.850666

frontiers | Frontiers in Plant Science

ORIGINAL RESEARCH
published: 25 April 2022
doi: 10.3389/fpls.2022.850666

Weed Identification by Single-Stage and Two-Stage Neural Networks: A Study on the Impact of Image Resizers and Weights Optimization Algorithms

Muhammad Hammad Saleem[1], *Kesini Krishnan Velayudhan*[1], *Johan Potgieter*[2] *and Khalid Mahmood Arif*[1]*

[1] Department of Mechanical and Electrical Engineering, School of Food and Advanced Technology, Massey University, Auckland, New Zealand, [2] Massey AgriFood Digital Lab, Massey University, Palmerston North, New Zealand

OPEN ACCESS

Edited by:
Dirk Walther,
Max Planck Institute of Molecular Plant Physiology, Germany

Reviewed by:
Borja Espejo-Garca,
Agricultural University of Athens, Greece
Mikhail Genaev,
Institute of Cytology and Genetics (RAS), Russia

***Correspondence:**
Khalid Mahmood Arif
k.arif@massey.ac.nz

Specialty section:
This article was submitted to Technical Advances in Plant Science, a section of the journal Frontiers in Plant Science

Received: 08 January 2022
Accepted: 11 March 2022
Published: 25 April 2022

Citation:
Saleem MH, Velayudhan KK, Potgieter J and Arif KM (2022) Weed Identification by Single-Stage and Two-Stage Neural Networks: A Study on the Impact of Image Resizers and Weights Optimization Algorithms. Front. Plant Sci. 13:850666. doi: 10.3389/fpls.2022.850666

The accurate identification of weeds is an essential step for a site-specific weed management system. In recent years, deep learning (DL) has got rapid advancements to perform complex agricultural tasks. The previous studies emphasized the evaluation of advanced training techniques or modifying the well-known DL models to improve the overall accuracy. In contrast, this research attempted to improve the mean average precision (mAP) for the detection and classification of eight classes of weeds by proposing a novel DL-based methodology. First, a comprehensive analysis of single-stage and two-stage neural networks including Single-shot MultiBox Detector (SSD), You look only Once (YOLO-v4), EfficientDet, CenterNet, RetinaNet, Faster Region-based Convolutional Neural Network (RCNN), and Region-based Fully Convolutional Network (RFCN), has been performed. Next, the effects of image resizing techniques along with four image interpolation methods have been studied. It led to the final stage of the research through optimization of the weights of the best-acquired model by initialization techniques, batch normalization, and DL optimization algorithms. The effectiveness of the proposed work is proven due to a high mAP of 93.44% and validated by the stratified k-fold cross-validation technique. It was 5.8% improved as compared to the results obtained by the default settings of the best-suited DL architecture (Faster RCNN ResNet-101). The presented pipeline would be a baseline study for the research community to explore several tasks such as real-time detection and reducing the computation/training time. All the relevant data including the annotated dataset, configuration files, and inference graph of the final model are provided with this article. Furthermore, the selection of the DeepWeeds dataset shows the robustness/practicality of the study because it contains images collected in a real/complex agricultural environment. Therefore, this research would be a considerable step toward an efficient and automatic weed control system.

Keywords: deep learning, convolutional neural network, weed detection, optimization algorithms, transfer learning

INTRODUCTION

With the fast-growing global population, food demand is expected to increase up to 70% by 2050 (Caldera and Breyer, 2019). Therefore, various challenges in the agricultural sector have been addressed by the research community to get smart and intelligent solutions. Among various agricultural problems, weeds are a serious threat to crop yield that causes economic loss (Ahmad et al., 2021). An effective way to manage the weed is to use herbicide spray specifically in the field that contains the weeds. Accurate and precise detection of weeds is important to successfully deploy the weed management system (Hasan et al., 2021). This agricultural task is time-consuming and requires a great amount of human and machine resources. Furthermore, fast and automatic identification of weeds is essential to reduce excessive/unrequired application of a chemical spray that produces adverse effects on human beings and ecosystems (Lottes et al., 2020).

After the introduction of the AlexNet model in 2012 (Krizhevsky et al., 2012), deep learning (DL) has recognized its ability to detect, classify, and localize several objects quickly. The object identification tasks are performed in controlled/laboratory and uncontrolled/real environments. Similarly, the agricultural field of research is being accelerated by leveraging various developments in DL. The research community is extensively focusing on agricultural tasks including fruit harvesting/recognition (Fu et al., 2021; Gai et al., 2021), plant recognition (Quiroz and Alférez, 2020; Bisen, 2021), identification of crop water stress (Chandel et al., 2021), land cover classification (Saleem et al., 2021), and plant disease detection (Priyadharshini et al., 2019; Saleem et al., 2019; Uguz and Uysal, 2021) by investigating DL-based techniques. Similarly, recent advances in DL have encouraged researchers to address the classification and detection of weeds in several plant species (Hasan et al., 2021).

On the other hand, after rapid developments in DL, still, the robustness of the DL-based solutions is a critical research question among the scientific community. There are various aspects to realize the strength of DL, and the environment of the collected dataset images is one of the important factors. It is commonly observed that DL architectures provide higher accuracy on the images collected in controlled or with plain/single background compared to those which were collected in a real environment. This is due to the presence of various unessential or background elements/objects that could have characteristics similar to the required objects. Furthermore, occlusion is another aspect of reducing or degrading the performance of the DL models. Therefore, agricultural researchers and data scientists started collecting images for the datasets in real agricultural environments. To address the concerns described above, a publicly available dataset called DeepWeeds (Olsen et al., 2019) has been used throughout this research, which contains various characteristics of the real agricultural environment.

A summary of prominent studies regarding the identification of weeds by various methodologies related to DL is presented in **Table 1**. It can be concluded that previous studies focused on DL-based weed detection in four ways: evaluation of transfer learning techniques, investigation of the performance of state-of-the-art DL models, integration of DL models with other image processing-based/traditional machine learning methods, and modification of the well-known DL architectures. To the best of the authors' knowledge, none of the previous articles has provided a comprehensive study of weed detection by performing an in-depth analysis of single- and/or two-stage DL-based object detectors along with an extensive investigation of various aspects of DL in terms of image resizers, image interpolation, weight initialization, batch normalization, and optimization methods. The major contributions of this study are: (1) a novel DL-based methodology is presented to identify the weeds by analyzing and evaluating the performance of various single-stage and two-stage neural networks; (2) also, the effects of various image resizing techniques are discussed. Moreover, weights of the best-obtained neural network are optimized with initialization method, batch normalization, and optimization algorithms; their effects on the training and testing datasets are also thoroughly studied; (3) the optimized/modified DL approach improved the mean average precision (mAP) with a significant margin as compared to the default settings; attained enhanced average precision in individual classes; the presented approach can be adapted to other agricultural operations due to a high mAP for weed detection; (4) an in-depth analysis of the best-obtained DL architecture is performed; it has provided a strong basis of the future research to propose a modified DL model for further enhancing the research on weed detection; (5) the trained weights of various DL models can also be used as transfer learning for other weed-related datasets. Moreover, the proposed methodology can be treated as an earlier step before modifying the hidden layers of neural networks for other agricultural applications.

MATERIALS AND METHODS

This article presents a DL-based approach to detect and classify eight types of weeds. First, a publicly available dataset called the DeepWeeds dataset is selected, which covers different aspects of a real agricultural environment. The proposed method comprises four steps. The first step is the analysis of various single-stage and two-stage object detectors and the best-suited DL model which attained the highest mAP. It led to the empirical evaluation of image resizing techniques like aspect ratio and fixed shape resizers by using the image interpolation methods including bilinear, bicubic, area, and nearest neighbor. Then, an attempt was made to optimize the weights of the DL model in two stages. First, the parameters of the weight initialization methods, such as the truncated normal, variance scaling, and random normal techniques, were studied. Later, the effects of batch normalization were studied. Finally, adaptive DL optimization techniques including Adam and RMSProp were applied to further enhance the performance of the best-obtained DL architecture as presented for other agricultural applications (Saleem et al., 2020a,b); their hyperparameters were tuned with the random search method. The final mAP was compared with

TABLE 1 | Summary of research articles related to weed detection by deep learning (DL) (divided in terms of novelty and research ideas of the work).

Novelty/research ideas	DL models	Number of classes	Dataset conditions	Best model performance	References
Investigation of DL models for the identification of weeds	AlexNet, GoogLeNet, VGGNet, DetectNet	Three	Various surface condition regimes.	DetectNet F1-score = 0.9843	Yu et al., 2019a
DL architectures were leveraged for weed detection and classification	DetectNet, GoogLeNet, and VGGNet	Three	Different stages and densities of growth	F1-score by DetectNet > 0.99	Yu et al., 2019b,c
Speed-optimized CNN models were proposed	CNN model	Two	Images were taken with a field robot in a real environment.	A speed-up factor of 31	Knoll et al., 2019
DL model used with color index-based segmentation	CenterNet	One	Different illumination conditions, backgrounds, and growth stages	F1-score by the CenterNet model = 0.953	Jin et al., 2021
A tiny version of the YOLO model was proposed to reduce the computation time	Modified tiny YOLO-v3, YOLO-v3-tiny	Two	Synthetic images were generated	Mean average precision: 0.829	Gao et al., 2020
Various factors to develop weed identification system along with the significance of transfer learning	AlexNet, VGG-F, VGG-VD-16, Inception-v1, ResNet-50, ResNet-101	Two	A robotic platform was used to take images on the field.	Accuracy by ResNet-101: 97.1+/-0.1%	Kounalakis et al., 2019
Two DL detectors were used through a UAV	Faster RCNN and SSD	Six	Images were taken by a camera mounted on a UAV	Mean IoU by Faster RCNN: 0.85	Veeranampalayam Sivakumar et al., 2020
An improved DL model was proposed	Proposed Faster RCNN, KNN, SVM, and YOLO-v3	Two	A camera mounted on a UAV in two agricultural fields	Overall average identification accuracy: 94.7%	Khan et al., 2021
A CNN model was optimized for real-time weed recognition	ResNet-18	Six	Dataset images were collected by a UAV	Overall accuracy: 94%	De Camargo et al., 2021
Three ML and DL-based methods were used and compared	SVM, YOLO-v3, and Mask R-CNN	Two	A multispectral camera mounted on a drone was used	F1-score by YOLO and RCNN models: 94%	Osorio et al., 2020
DL-based classification and detection models were used	VGG-16, ResNet-50, Inception-v3, YOLO-v3	Four	Images were collected in a real field environment	mAP: 54.3%	Ahmad et al., 2021
A graph CNN-based model was proposed to detect weeds	GCN-ResNet-101, AlexNet, ResNet-101, VGG-16	Four	The Weeds were collected in three crops and a fourth was obtained by combining the three datasets.	Average recognition accuracy: 98.15%	Jiang et al., 2020
A combination of DL and ML methods was considered	Xception, Inception-ResNet, VGNets, MobileNet, DenseNet, SVM, XGBoost, and Logistic Regression	Two	The dataset was collected under variable soil, color and illumination conditions.	F1-score: 99.29%	Espejo-Garcia et al., 2020

the one obtained by default settings to show the effectiveness of this research. The obtained mAP was validated by the stratified k-fold cross-validation method. An overall methodology is also presented in **Figure 1**.

Selection of the Dataset

The main criteria for selecting the dataset were the images should be collected in a real agricultural environment considering various features of the actual field. These characteristics include natural background, occlusion, rotation, geographical/seasonal changes in plants, and variable lighting conditions. As the DeepWeeds dataset (Olsen et al., 2019) had all these characteristics, it was selected for this research. These conditions were important to consider because a higher accuracy attained on this kind of dynamic dataset would prove the robustness of the proposed work. The dataset contains 17,509 images divided into eight classes of weeds, including one negative class that has non focused plants, and the images were collected in Northern Australia.

Dataset Division and Annotation

The DeepWeeds dataset was divided into three sub-datasets: training (70%), validation (20%), and testing (10%). For models like Single-shot Multibox Detector (SSD), RetinaNet, Faster Region-based Convolutional Neural Network (RCNN), Region-based Fully Convolutional Networks (RFCN), CenterNet, and EfficientDet in TensorFlow object detection Application Programming Interface (API), the dataset images were annotated using an open-source image annotation tool called LabelImg. The bounding box coordinates were obtained in Xmin, Xmax, Ymin, and Ymax. The annotations were saved in XML format, which was converted to CSV format and later to TF records (Saleem et al., 2020a). Unlike TensorFlow models, images annotated in XML format were then converted to TXT format to train the YOLO-v4 model. To visualize the detected results within

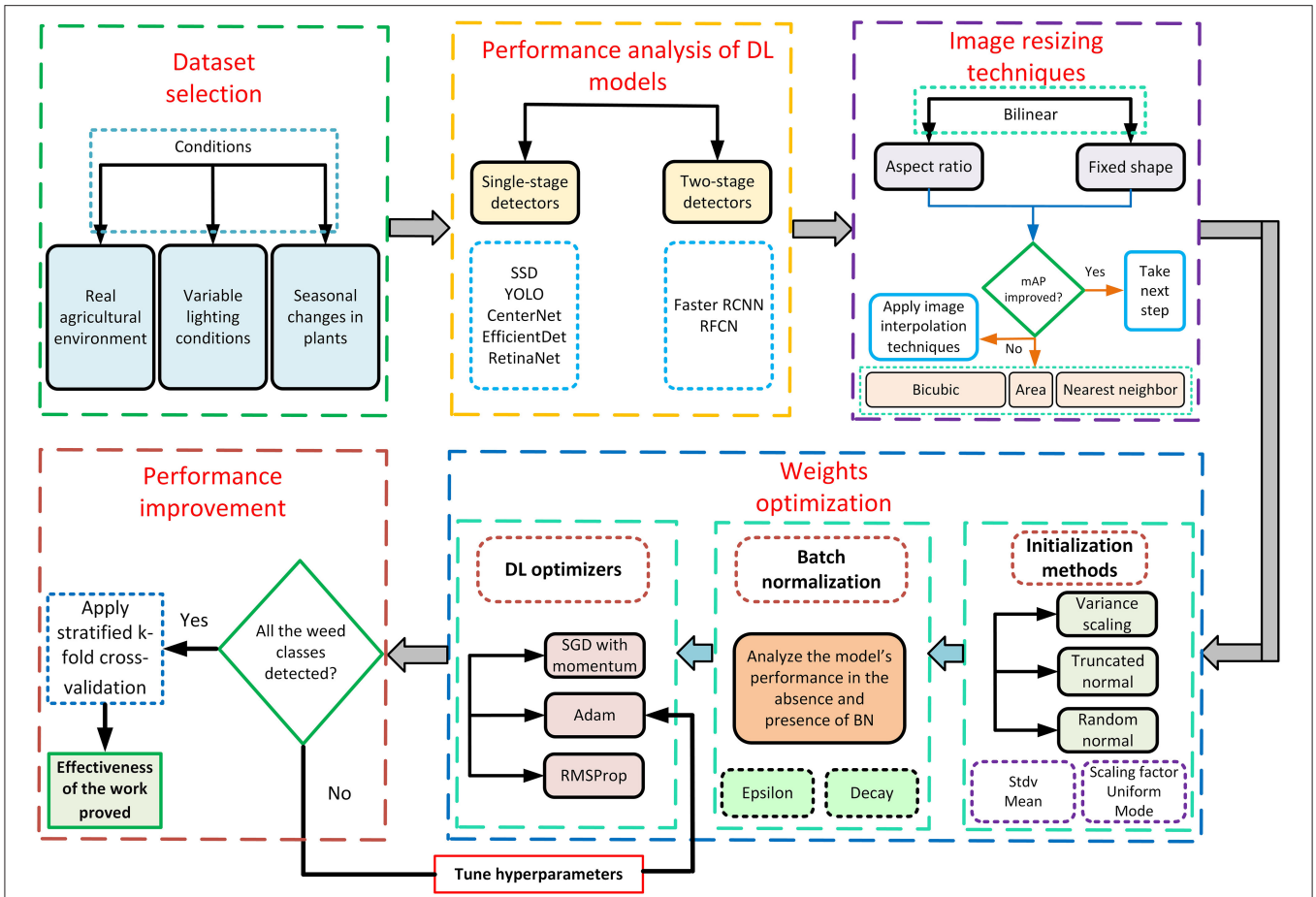


FIGURE 1 | Framework of this research.

a bounding box, a shortened name of each class was set to label the images before training. For example, chinee apple was replaced by C_App, lantana was replaced by Lntna, prickly acacia was replaced by P_acacia, parthenium with P_nium, parkinsonia with P_sonia, rubber vine with R_vine, siam weed with S_weed, snakeweed with Snk_wd, and negative with Ngtv. A sample of annotated images of each class is presented in **Figure 2**.

DL-Based Object Detectors

In this article, the performance of various versions of DL models integrated with different feature extractors or backbone models were analyzed on the selected dataset. These DL meta-architectures are divided into two categories: single-stage and two-stage neural networks. Among single-stage detectors, state-of-the-art models like SSD (Liu et al., 2015), CenterNet (Duan et al., 2019), EfficientDet (Tan et al., 2019), RetinaNet (Lin et al., 2017), and YOLO-v4 (Bochkovskiy et al., 2020) were included. These models were trained using TensorFlow (1 and 2) object detection API and the YOLO/Darknet neural network framework. The two-stage DL models were also trained and tested using TensorFlow Object Detection API 1. These networks generally contain the first stage of the region proposal by Region Proposal Network (RPN). While the second stage refines the classification and localization of the proposals. The most prominent among them were the Faster RCNN (Ren et al., 2015) and RFCN (Dai et al., 2016) models.

Image Resizing Techniques

The second step of the proposed method was the evaluation of the best-obtained DL object detector using image resizing techniques along with image interpolation methods. In this regard, the input images were resized to a fixed shape or by using an aspect ratio having minimum and maximum image dimensions in pixels. For example, Faster RCNN used a shorter pixel value of 600 and a longer one equal to 1,000 pixels as default values. Furthermore, these image resizers were used with interpolation techniques including bilinear (bilinear interpolation), bicubic (cubic interpolation), area, and nearest neighbor (multivariate interpolation for multiple dimensions).

Weight Initializers

The approach proposed in this article extensively considers weight optimization in three ways. First, the effects of initialization methods were analyzed depending on the type

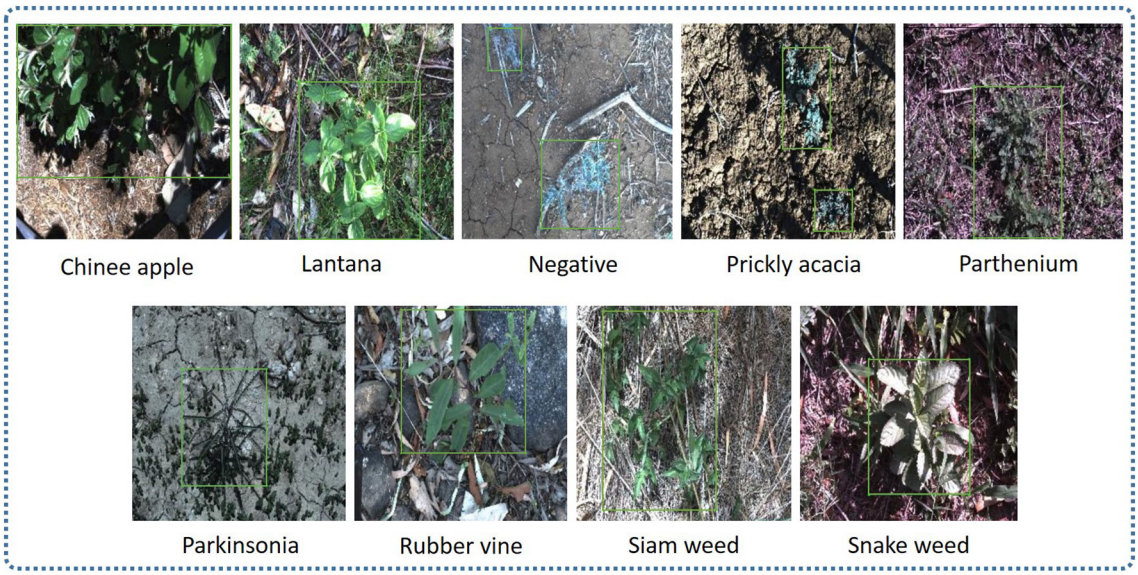


FIGURE 2 | Sample of the annotated dataset for each class.

of neural network layer/activation function. Three initialization methods include truncated normal, variance scaling, and random normal initializers. The truncated normal creates a tensor having a truncated normal distribution, which is useful to avoid dead neurons due to ReLU activation functions. It discards and redraws values more than two SD from the mean. It is the most recommended weight initialization technique for neural network-based DL models.

When the ReLU activation function came after the Sigmoid function, it was proven to successfully solve the problem of vanishing gradients. Then a weight initialization technique was proposed which balances the variance of the output layer with the input layer (He et al., 2015) and is known as He initialization. In the TensorFlow Object Detection API, the He initialization is named as the variance scaling initializer. The last initializer is a random normal initializer that is used to generate tensors with a normal distribution.

Batch Normalization

Batch normalization was introduced to solve the problem of internal covariate shift due to the change in the distribution of the input of the neural network layer with the change in the parameters of the previous layers (Ioffe and Szegedy, 2015). The use of BN increases the training speed to get the convergence of the model with a high learning rate.

Deep Learning Optimization Techniques

The next step of the research was weight optimization using DL optimizers. Their hyperparameters were tuned with the help of the random search method. Three optimizers were used for this purpose, namely, Stochastic gradient descent (SGD) with momentum, Root mean square propagation (RMSProp), and Adaptive moment estimation (Adam). SGD (with momentum) is one of the most commonly used DL optimizers to train DL architectures for various applications (Saleem et al., 2020a) due to its fast convergence, which is a result of the inclusion of an exponentially weighted average of weights and bias gradients (Ruder, 2016). On the other hand, RMSProp limits the oscillations generated during the training by considering the square of gradients of weights and biases. Furthermore, it allows the algorithm to consider a larger learning rate (Hinton et al., 2012). Adam optimizer is the combination of RMSProp and momentum optimizers. It includes an exponentially decaying average of the previous gradient with squared gradients (Kingma and Ba, 2014).

Stratified k-Fold Cross-Validation Technique

The DeepWeeds dataset has the class imbalance problem, for example, the negative/non-weed class has a significantly higher number of images as compared to all eight classes of weed. Therefore, a stratified k-fold cross-validation technique was selected to validate the final results. This method avoids biasness while creating the folds of training/testing dataset and allows to maintain the same proportion of each class sample in each fold, as in the initial distribution (He and Ma, 2013). It was made sure that the testing images in each fold were not the same.

Software and Training Specifications

The DL meta-architectures were trained and tested using TensorFlow object detection API 1, 2, and YOLO/Darknet neural network framework. All experiments were carried out using a Graphical Processing Unit (NVIDIA GeForce GTX 1080 Ti) with specifications: 11 GB memory, 1,582 MHz boost clock, 3,584 CUDA cores, and 484 GB/sec memory bandwidth. CuDNN library was imported to increase the training speed.

TABLE 2 | Hyperparameters of deep learning optimization algorithms with their respective DL architectures.

DL models	DL optimizers	Hyperparameters
Yolo-v4	SGD with momentum	learning rate = 1 x 10^{-3}, momentum = 0.9
RetinaNet		learning rate = 3 x 10^{-4}, momentum = 0.9
EfficientDet		learning rate = 2 x 10^{-4}, momentum = 0.9
RFCN ResNet-101		learning rate = 4 x 10^{-4}, momentum = 0.9
Faster RCNN Inception-v2		learning rate = 2 x 10^{-4}, momentum = 0.9
Faster RCNN ResNet-50		learning rate = 3 x 10^{-4}, momentum = 0.9
SSD MobileNet	RMSProp	learning rate = 2 x 10^{-3}, rho = 0.9, momentum = 0.9, epsilon = 1.0 x 10^{-2}
SSD Inception-v2		learning rate = 2 x 10^{-4}, rho = 0.9, momentum = 0.9, epsilon = 1.0 x 10^{-4}
CenterNet ResNet-50	Adam	learning rate = 1 x 10^{-3}, epsilon = 1 x 10^{-7}
Faster RCNN ResNet-101	SGD with momentum	learning rate = 3 x 10^{-4}, momentum = 0.9
	Adam	learning rate = 1 x 10^{-5}, epsilon = 1 x 10^{-2}
	RMSProp	learning rate = 3 x 10^{-4}, rho = 0.9, momentum = 0.9, epsilon = 1.0

To leverage transfer learning, the pre-trained weights on the COCO dataset were used. Depending on the DL architecture and GPU limitations, batch sizes equal to 2, 4, 6, and 8 were the most reasonable values to minimize the trade-off between mAP and training time (Masters and Luschi, 2018). The learning rate and the values of other hyperparameters were selected by the random search technique (Bergstra and Bengio, 2012) as presented in **Table 2**.

RESULTS AND DISCUSSION

This article aims to identify and localize eight classes of weeds using DL architectures. In this regard, seven DL architectures were trained with different feature extractors/backbone models. The performance of these architectures is evaluated in terms of mAP, which is a commonly used performance metric for object detection tasks. Equation (1) presents the formula to evaluate mAP.

$$mAP = \frac{\sum_{i=1}^{n} AP_i}{n} \quad (1)$$

where AP is the average precision calculated for each class and accessed by the 11-point interpolated AP method and n is the number of classes. The AP is defined as the average precision across all unique recall levels. Therefore, the precision at various recall values is first evaluated. Then, interpolated precision is calculated as the maximum precision for a certain recall level. Further details of this performance metric can be found in (Saleem et al., 2020a).

This section is divided into two steps. First, the weed detection results obtained by the single- and two-stage DL architectures are provided along with their class-wise performance analysis and loss plots. Secondly, the effects of various image resizing techniques on the best-obtained DL model are presented. Furthermore, the weights were optimized by leveraging initialization techniques along with batch normalization and DL optimizers. Finally, a significant improvement in the mAP of the optimized DL architecture is discussed compared to its default settings to demonstrate the effectiveness of the proposed approach.

Step 1: Selection of the Best-Suited DL Architecture

Performance of Single-Stage Neural Networks

YOLO-v4

This DL architecture has the backbone model CSPDarknet-53 having input image size 608 x 608. The SGD with momentum optimizer was used to train the model. Various batch sizes were tested and eight was found to be the most feasible to reduce the trade-off between accuracy and training time as minimum as possible. The plot in **Figure 3A** shows that the model started to converge after 48K iterations, and the model training took around 12 h. The final average loss value was found to be 2.83%. A few classes of weeds were successfully identified, including parthenium, rubber vine, and siam weed, with an average precision of more than 90%. Therefore, their distinct characteristics were successfully extracted. None of the classes attained <50% AP which shows the significance of this model to detect several classes of weed. An example of the three classes which achieved the highest AP is presented in **Figure 3B**. Furthermore, some of the images of classes such as lantana and snakeweed were undetected, as shown in **Figure 3C**. The mAP of all classes was calculated to be 79.68% as shown in **Table 3**. Each prediction box is related to the class label with a confidence score of 0 to 100% (0 to 1).

SSD

A Single-shot MultiBox detector was trained with different backbone models to extract the features of weed classes. The feature extractors like Inception v2 and MobileNet from TensorFlow 1 API were considered. The fixed input image resizer was applied with 300 × 300 dimensions. The model was trained with an RMSProp optimizer, as the momentum optimizer was unable to converge the training. It took around 11 h to complete the training in 70K steps for the Inception model, and the batch size was equal to 8. However, the MobileNet model took around

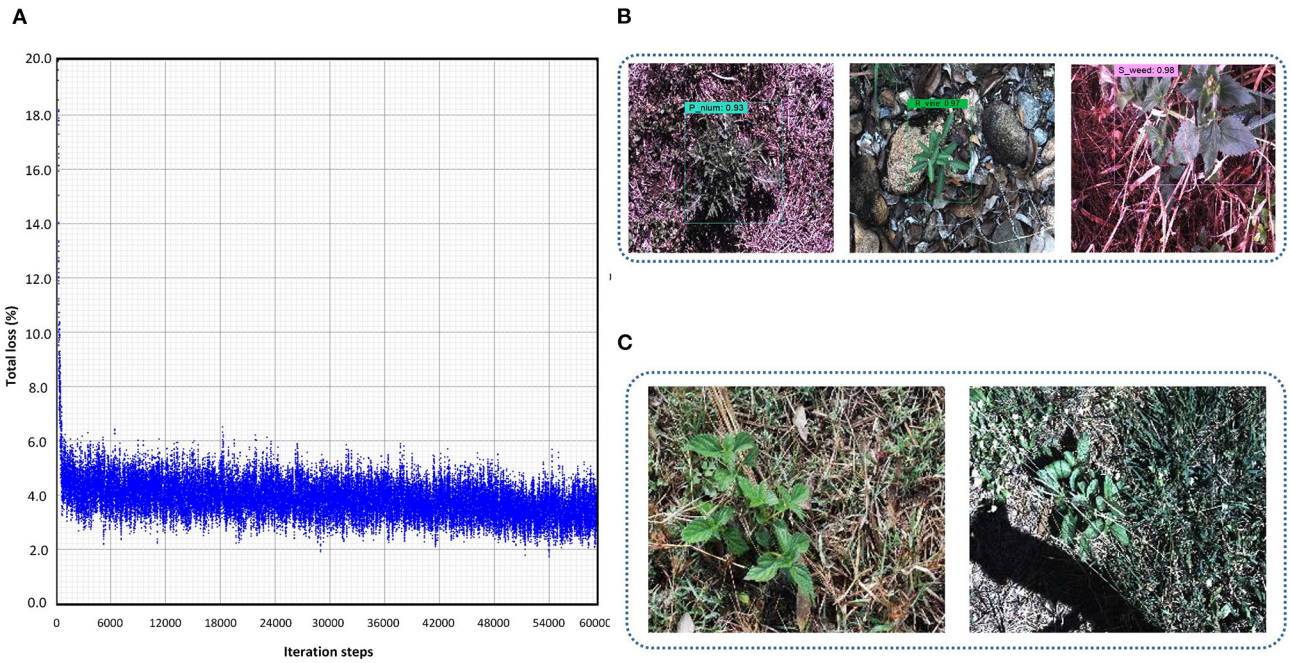


FIGURE 3 | Performance of the You look only Once (YOLO)-v4 model: **(A)** loss plot; **(B)** true positives for parthenium, rubber vine and siam weed; **(C)** examples of undetected images for lantana and snake weed classes.

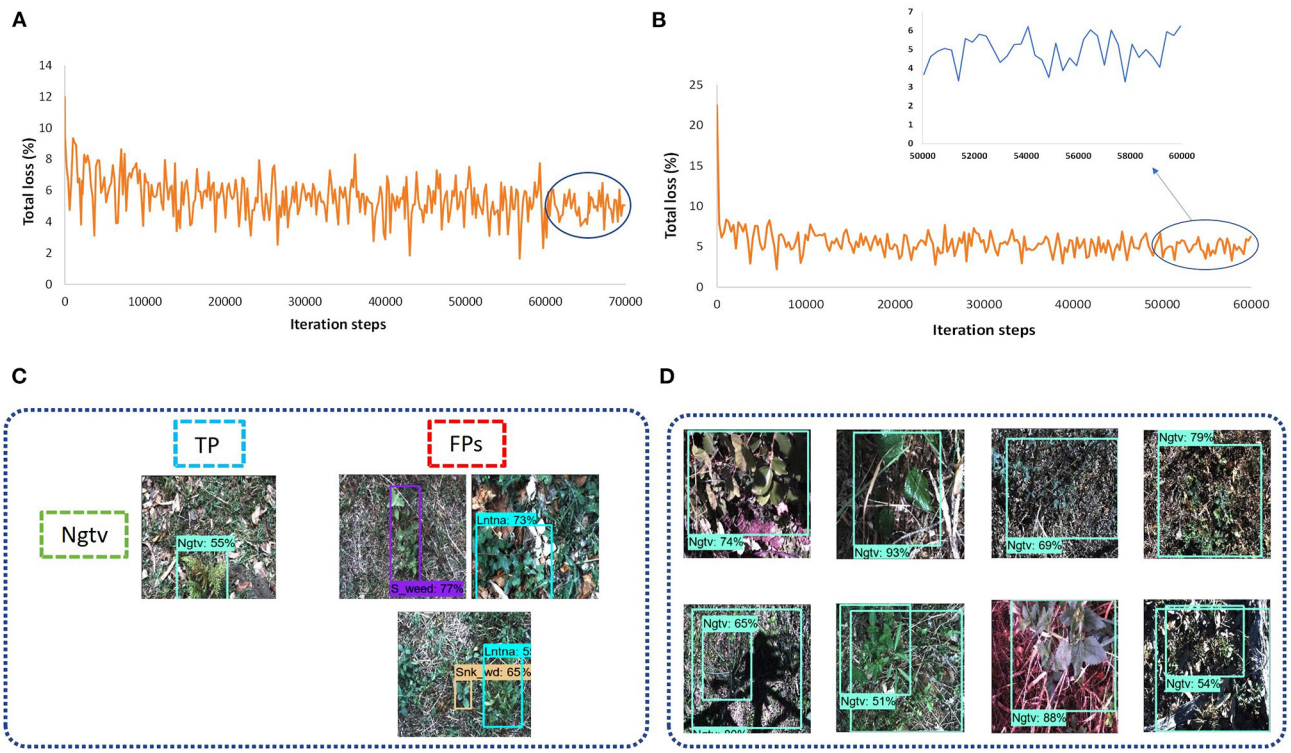


FIGURE 4 | Performance of the single-shot multibox detector (SSD) architecture: **(A)** total loss with the Inception model; **(B)** total loss with the MobileNet model; **(C)** examples of a false-positive result for the negative class with the Inception-v2 model; **(D)** example of false positives for the eight classes of weeds with the MobileNet model. TP: true positive, FP: false positive.

TABLE 3 | Summary of the weed detection results of the DL single-stage and two-stage object detectors in terms of the average precision (in %) of each class.

Annotated weed and negative classes	DL architectures with backbone models									
	Single-stage networks						Two-stage networks			
	Yolo-v4 CSPDarknet-53	SSD			EfficientDet EfficientNet	CenterNet ResNet-50	RFCN ResNet-101	Faster RCNN		
		Inception-v2	MobileNet	ResNet-50 (RetinaNet)				Inception-v2	ResNet-50	ResNet-101
C_App	67.4	26.25	18.83	45.31	43.97	26.34	34.29	100	98.21	99.87
Lntna	66.61	62.22	31.65	9.09	28.79	9.09	100	96.83	99.45	82.46
P_acacia	73.87	34.45	0.75	9.09	9.67	1.82	56.5	28.64	94.08	70.06
P_nium	93.48	54.16	26.36	17.88	33.84	23.85	38.83	99.94	99.94	99.33
P_sonia	79.51	53.93	30.92	17.05	44.23	35.77	99.7	99.24	99.89	88.85
R_vine	96.33	60.44	76.99	27.27	44.18	35.88	92.9	99.77	100	99.84
S_weed	98.6	66.06	26.35	54.55	63.29	53.31	41.61	82.49	100	99.85
Snk_wd	58.19	62.4	21.36	14.91	34.79	33.57	0.55	4.17	15.38	86.17
Ngtv	83.17	13.2	81.72	0.11	26.57	26.61	31.18	51.28	78.13	62.35
mAP (%)	79.68	48.12	34.99	21.69	36.59	27.36	55.06	73.59	87.23	**87.64**

Bold values shows the highest mAP to select the best DL architecture.

5 h to converge the training in 60K steps, which was the fastest among all other models due to its fewer parameters (Huang et al., 2016). The total loss was fluctuated between 4 to 6% in the case of the Inception-v2 model (**Figure 4A**) while it was 3 to 6% for the MobileNet model (**Figure 4B**). From the results, it can be concluded that none of the weed classes was able to achieve an AP of more than 90%, with the SSD model trained with Inception and MobileNet feature extractors. However, the siam weed class achieved the highest AP of 66.06% with the Inception model, and the negative class achieved the lowest AP of 13.2%. The reason for the false detection of the negative class was confusion with three classes, including snakeweed, siam weed, and lantana, as presented in **Figure 4C**. The negative class worked well with the MobileNet model, while the prickly acacia was almost undetected, as its AP was only 0.75%. All weed classes were confused with the negative class when the feature extractor was MobileNet as shown in **Figure 4D**. It resulted in a lower mAP of approximately 35%, as shown in **Table 3**.

RetinaNet

In this research, three DL object detectors from TensorFlow object detection API 2 were also tested. The first model was the RetinaNet, which had an SSD model as a base architecture, and ResNet-50 was used as a feature extractor. Although other versions of ResNet (with 101 and 152 layers) were also available in the API, due to GPU memory limitations, only ResNet-50 was feasible to train and test on the DeepWeeds dataset. An input image fixed shape resizer of 640 × 640 dimensions was used with the SGD optimizer. The batch size equal to 4 was found to be a reasonable value. With all the described settings, the model took around 14 h. The loss plot to understand the training performance of the model is presented in **Figure 5A**. In almost 60K iterations, the model settled to its final loss value with a small fluctuation between 0.55 and 0.75%. None of the classes achieved a satisfactory AP, and the siam weed class achieved the highest AP of 54.55% among all other classes. Three classes including lantana, prickly acacia, and negative achieved the lowest AP of 9.09, 9.09, and 0.11% respectively. Most of the testing images belonging to the negative class did not detect and the remaining images resulted in parthenium and siam weed as shown in **Figure 5B**.

EfficientDet

Another single-stage DL object detector was utilized for this study named EfficientDet. This model used the aspect ratio resizer technique with dimension 512 × 512. The other versions of the model couldn't be trained due to GPU limitations. The optimum batch size was 4 with a momentum optimizer, and this setting required 11.5 h to get convergence in the training to 70K iterations. According to the loss plot presented in **Figure 6A**, the model received a final loss ranging between 0.25 and 0.45%. The model also detected the siam weed with the highest AP of 63.29%. However, the prickly acacia got the lowest average precision among all seven other classes of weeds. Just like the other single-stage detectors, this DL model also gave false positives for the seven classes in terms of negative class as shown in **Figure 6B**. This model could also not detect any class of weeds with a higher AP.

CenterNet

The last model among single-stage DL object detectors was CenterNet which had various versions in TensorFlow API including Hourglass, ResNet-101, and MobileNet-v2. Among them, only 50 layered ResNet feature extractor was able to detect some of the weed images with 512 x 512. However, most of the images of the eight classes of weeds could not be detected well. Moreover, the SGD with momentum optimizer failed to detect the testing images with the CenterNet model. Therefore, an Adam optimizer was used to train the model, and the batch size was set to 6. The model got converged after 60k iterations and it took around 12.5 h to train the model. The final loss ranged from approximately 1.5 to 2.5%, as shown in **Figure 7A**. This model

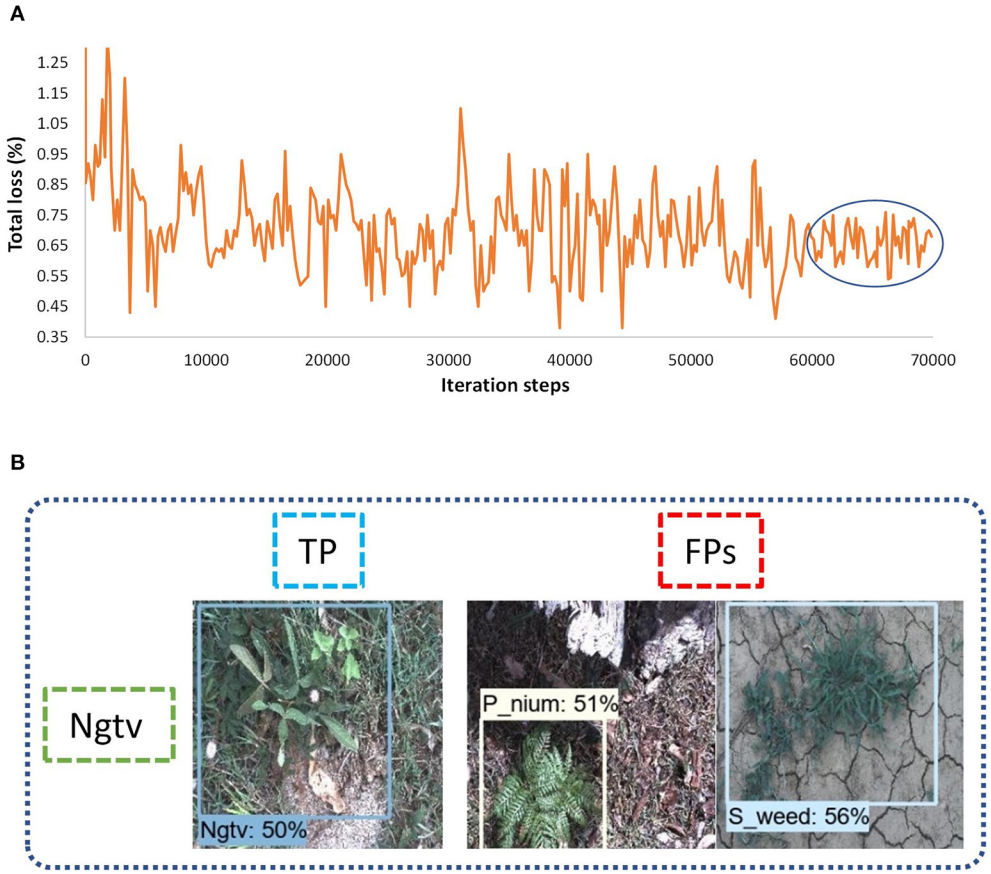


FIGURE 5 | Performance of RetinaNet: **(A)** total loss plot; **(B)** examples of false-positive results for the negative class.

also achieved more than 50% average precision for the siam weed class and almost failed to detect the prickly acacia class having only 1.82% AP. However, the chinee apple and the lantana were confused with snakeweed as presented in **Figure 7B**.

Performance of Two-Stage Networks

RFCN

This article also considers two-stage DL object detectors like RFCN and Faster RCNN. Both the RFCN and Faster RCNN models were trained to 60k iteration steps with input aspect ratio, minimum pixel size 600, and maximum 1,000. ResNet-101 was used as the backbone model and SGD with momentum optimizer was used to train the RFCN model with batch size 2. It took 10 h to get the convergence in 60K iterations. The total loss was reduced to almost 1.5% as shown in **Figure 8A**. The model was successful to detect three classes of weeds (lantana, parkinsonia, and rubber vine). However, classes such as negative and snakeweed were confused with the other classes, as shown in **Figures 8B,C**. Overall, five classes of weeds achieved an average precision of <50%.

Faster RCNN

At last, the Faster RCNN model was trained with several feature extractors. The backbone models that included Inception ResNet, ResNet, and Inception were available along with their trained weights on the COCO dataset in TensorFlow object detection API 1. Three models, including Inception-v2, ResNet-50, and ResNet-101, were able to train with the available graphics memory. The total training loss with the Inception-v2 model was settled at almost 1.5% (as shown in **Figure 9A**). However, models like ResNet-50 and ResNet-101 got their convergence having fluctuation between 0 and 1%, as shown in **Figures 9B,C**, respectively. Furthermore, it can also be observed that the Faster RCNN model with the ResNet-101 model converged earlier than the ResNet-50 model. All versions of the Faster RCNN model had an input with the aspect ratio image resizing technique having 600 minimum and 1,000 maximum pixel dimensions, and the batch size was set to 2. Furthermore, the momentum was used to optimize the weights in this step of the proposed study; the models were trained up to 60K iteration steps. Among these three backbone models, the Inception model trained in the shortest time of around 8.5 h. However, ResNet-50 and ResNet-101 required 9 and 10 h, respectively.

For concluding the detection results, the Inception model was successful to detect five classes of weed. However, the prickly acacia and snakeweed obtained a low average precision. A few examples of both classes with their false-negative results are

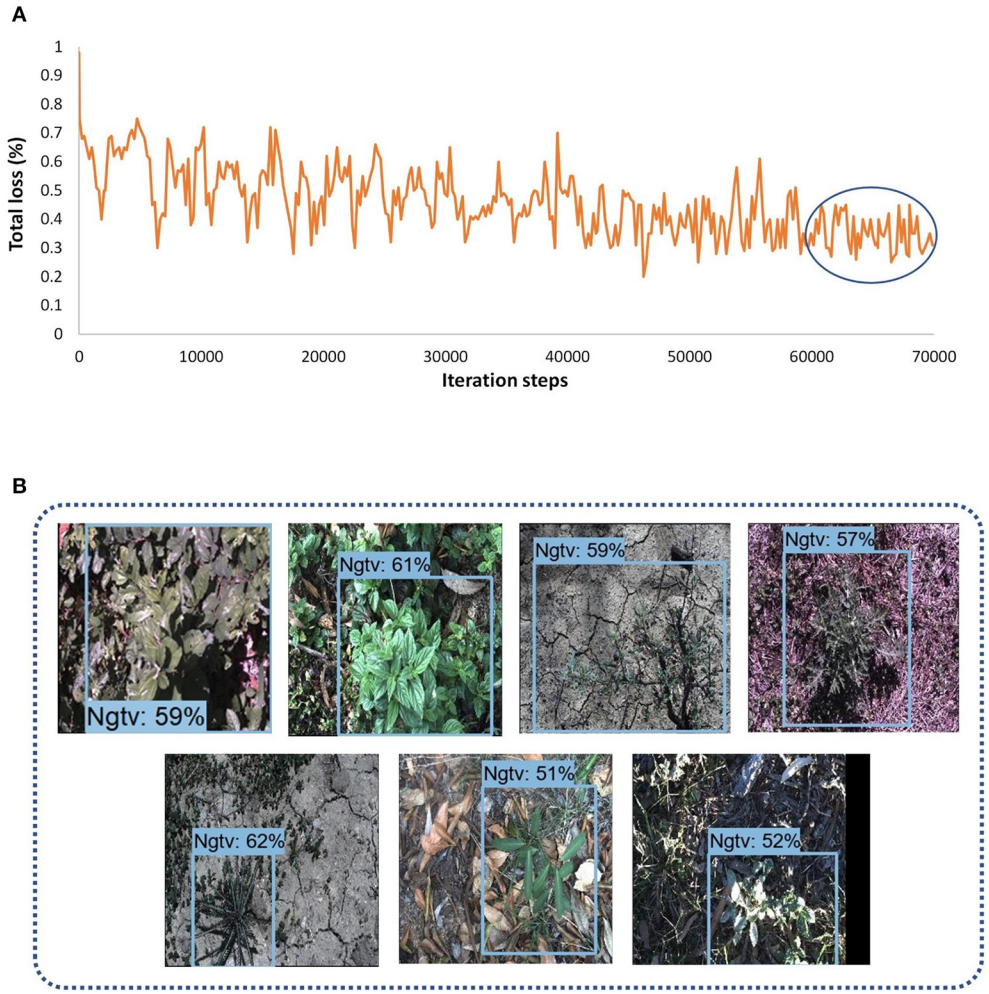


FIGURE 6 | Performance of the EfficientDet model; **(A)** total loss plots; **(B)** false positives of different classes with negative class.

shown in **Figures 9D,E**. The ResNet-50 model detected seven classes of weeds with high AP (more than 90%) as shown in **Figure 9F**. The negative classes also achieved an acceptable percentage of AP as shown in **Table 3**. However, the snakeweed class suffered from lower precision, as it was confused with the chinee apple, lantana, and siam weed classes.

Faster RCNN ResNet-101 was found to be the most suitable DL architecture for this study due to its highest mAP compared to all other DL architectures. This model succeeded to detect four classes of weed with more than 90% average precision; three classes achieved more than 80% AP. This model achieved more than 50% AP for negative; the prickly acacia achieved the lowest AP of almost 70% among all the other classes of weeds. Few samples of images detected by the Faster RCNN ResNet-101 model are presented in **Figure 9G**. From the results, it can be understood that the model was successful in extracting the unique features of several classes of weeds, but could not correctly extract the features of classes such as negative and prickly acacia. Therefore, the overall performance of this model was optimized in the second step of the proposed research. It further improved the mAP along with an enhancement in the average precision of individual classes.

Step-2: Optimization of the Faster RCNN ResNet-101 Model

The Faster RCNN model trained with the ResNet-101 feature extractor achieved the highest mAP. Therefore, the rest of the steps of the proposed work were applied to this DL architecture and the effects of image resizing techniques, weight initializers, batch normalization, and DL optimization algorithms were studied.

Effects of Image Resizing Techniques

This research evaluated the effects of two image resizing techniques, including fixed shape and aspect ratio, along with four image interpolation methods such as bilinear, bicubic, area,

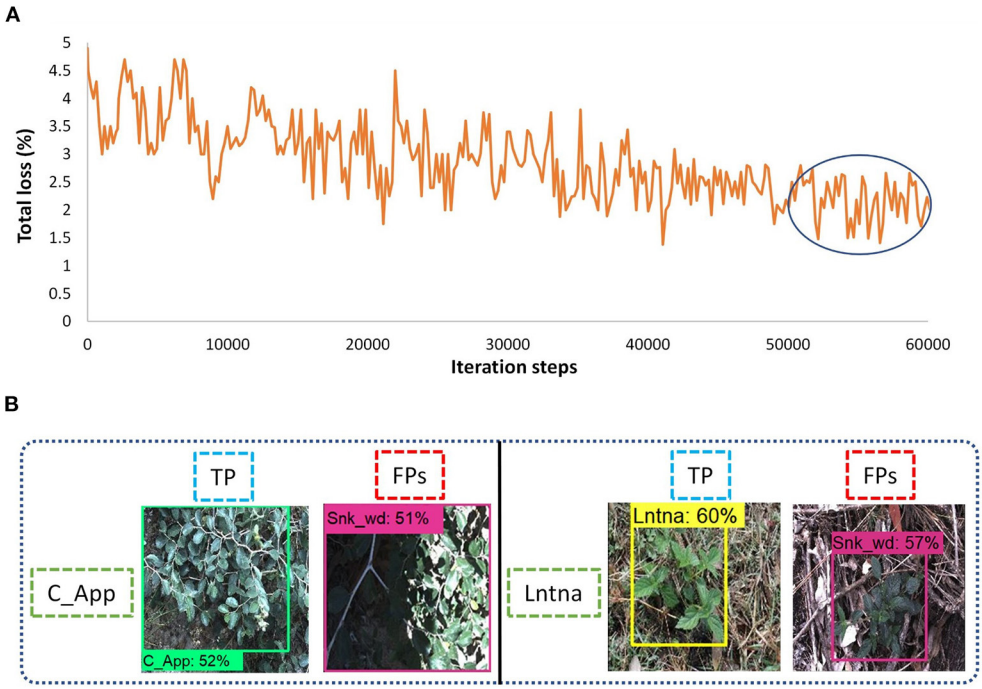


FIGURE 7 | Performance of the CenterNet model: **(A)** total loss; **(B)** false-positive results for the chinee apple, and lantana classes.

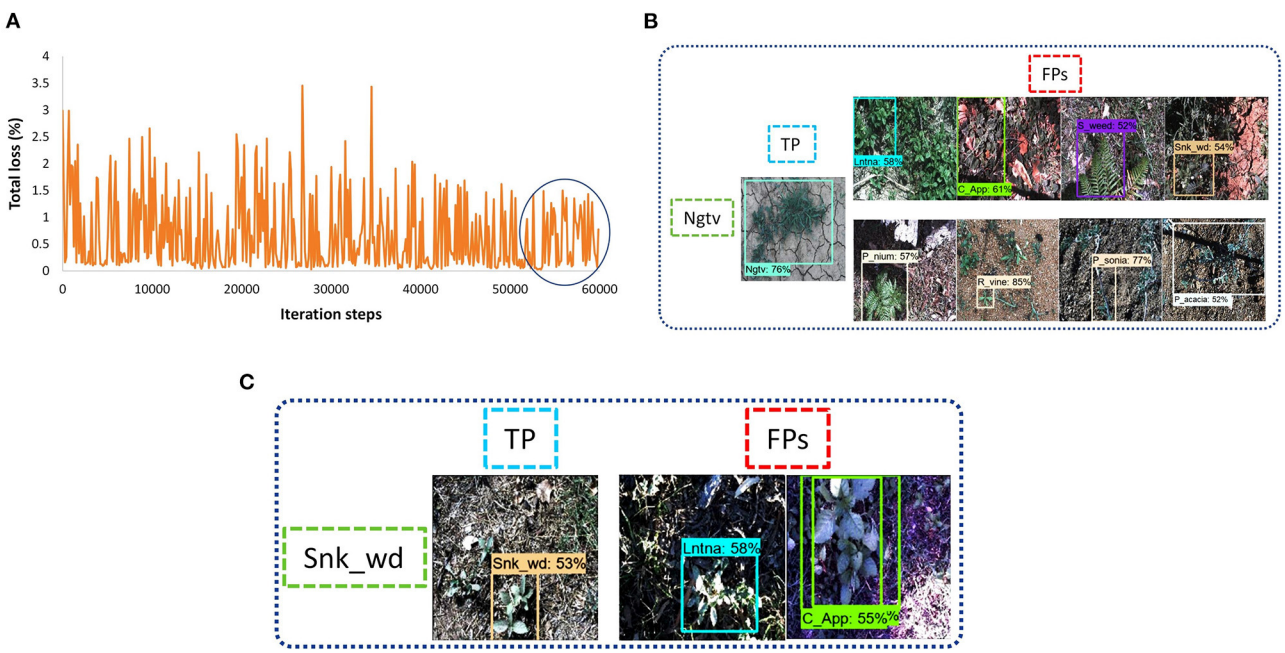


FIGURE 8 | Performance of the Region-based fully convolutional network (RFCN) model: **(A)** total loss plot; **(B)** false-positive results for the negative class; **(C)** false-positive results for the snakeweed class.

and nearest neighbor. First, the default settings of the Faster RCNN model were considered and tested using the model with the hyperparameters described in an earlier section. Later, the best image resizing method was found considering bilinear interpolation as the default method. It led to the application of three other interpolation methods. An overall mAP along

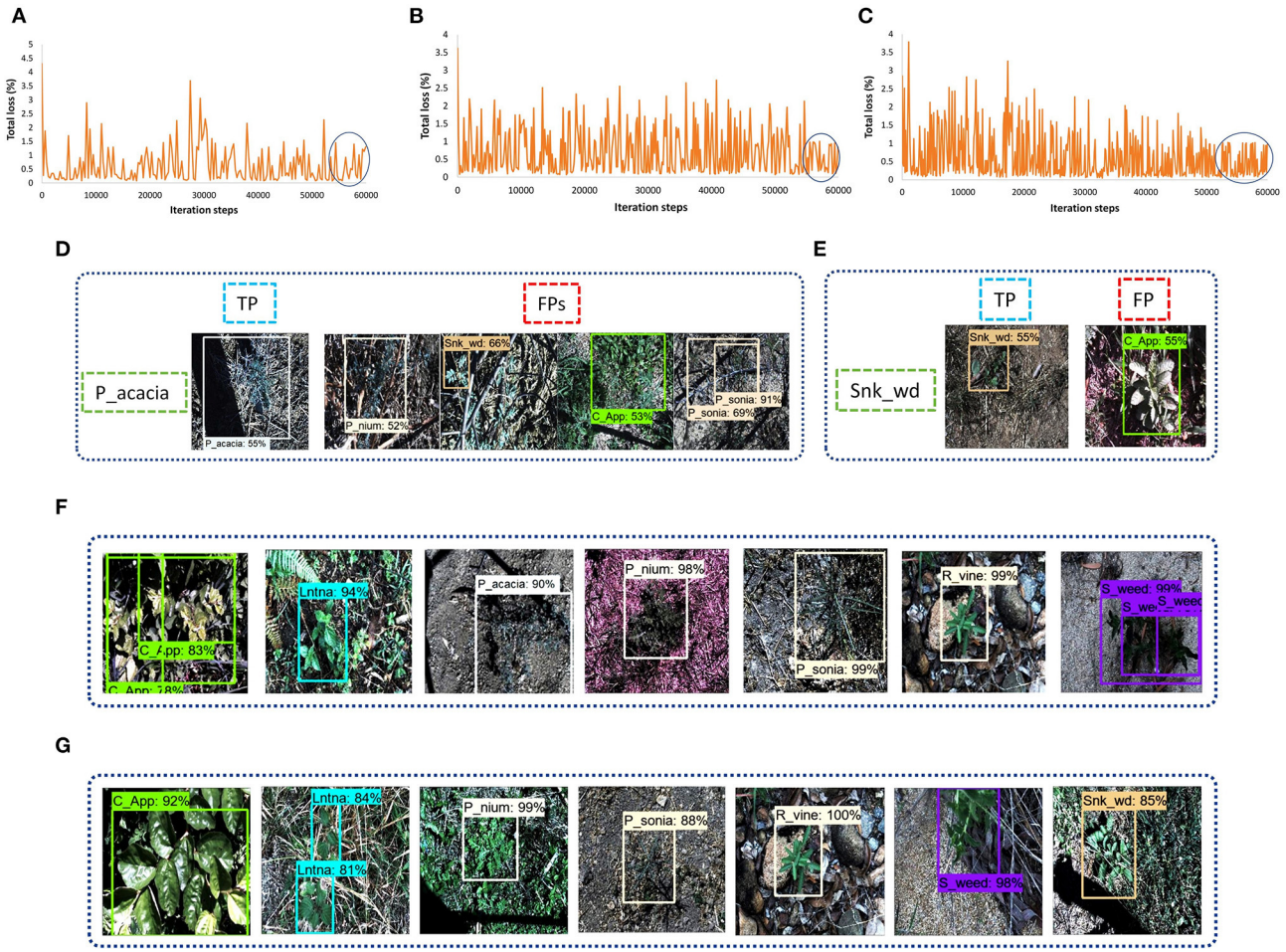


FIGURE 9 | Faster Region-base fully convolutional neural network (RCNN) performance with various versions of DL models: **(A)** total loss plot for Inception-v2: **(B)** total loss plot for ResNet-50; **(C)** total loss plot for ResNet-101; **(D)** false positives of prickly acacia with Inception-v2; **(E)** false positives of snake weed classes with Inception-v2; **(F)** true positives with ResNet-50; **(G)** true positives with ResNet-101.

with class-wise precision of each weed class was evaluated. Furthermore, various types of training losses are also presented to show the dependence of image resizers/interpolators on the performance of the DL model. The most suitable technique was selected for the next phase of the research. In this regard, the following observations are taken from this stage of the proposed methodology.

- In the Faster RCNN paper (Ren et al., 2015), the aspect ratio image resizer was selected as the default technique with minimum and maximum pixel dimensions equal to 600 and 1,000, respectively, and considered bilinear interpolation as the default method.
- The mAP of the default resizer was found to be 87.64%, as shown in **Table 3**.
- Although, the Faster RCNN model achieved good detection results. However, there was room to further enhance the performance of the neural network with other resizing methods.
- Later, the fixed-shape resizer method was applied. This method was first tested with bilinear interpolation, which provided a comparatively lower mAP. Furthermore, three interpolation methods like bicubic, area, and nearest neighbor with fixed image resizer attained lower mean average precision compared to the default technique, which was the aspect ratio with bilinear interpolation.
- Therefore, the aspect ratio resizer was selected as the best image resizing method for training the Faster RCNN ResNet-101 model. Here, the effectiveness of the aspect ratio resizing technique has also been validated through the experiments presented in this article, as it was also applied in the original Faster RCNN paper (Ren et al., 2015).
- The bilinear interpolation takes the closest 2 x 2 neighborhood of known pixel values and calculates the weighted average of 4 pixels to get the resultant interpolated value (Malik et al., 2017). Therefore, the pixels of weed images were interpolated to get sharper images to be provided as an input to the Faster RCNN model.

TABLE 4 | Summary of results and conclusions from each step of the proposed methodology.

Experiment/step of the analysis	DL models	Training details	Model assessment on training and testing datasets			Model performance analysis	Link to training code
		(Image resizers/interpolators/ initializers/optimizers)	Total loss (%)	Training time (h)	mAP (%)		
Training with default settings	Yolo-v4	FS (608 x 608)	2.83	12	79.68	Few of the weed classes were successfully identified	GitHub
	SSD Inception-v2	FS (300 x 300)	4–6	11	48.12	None of the weed classes was succeeded in achieving an AP of more than 90%	
	SSD MobileNet-v2	FS (300 x 300)	3–6	5	34.99	Fastest model convergence, but unsatisfactory testing outcomes	
	SSD ResNet-50 (RetinaNet)	FS (640 x 640)	0.55–0.75	14	21.69	Achieved the lowest mAP among all the DL models	
	EfficientDet EfficientNet	AR (min: 512, max: 512)	0.25–0.45	11.5	36.59	Eight classes attained AP of <50%	
	CenterNet ResNet-50	AR (min: 512, max: 512)	1.5–2.5	12.5	27.36	None of the classes achieved a satisfactory AP	
	RFCN ResNet-101	AR (min: 600, max: 1,000)	1.50	10	55.06	The model was successful to detect three classes of weeds	
	Faster RCNN Inception-v2	AR (min: 600, max: 1,000)	1.50	8.5	73.59	The model was successful to detect five classes of weed	
	Faster RCNN ResNet-50	AR (min: 600, max: 1,000)	0–1	9	87.23	Seven classes of weeds with high AP (more than 90%)	
	Faster RCNN ResNet-101	AR (min: 600, max: 1,000)	0–1	10	87.64	The most suitable DL architecture for this study due to its highest mean average precision compared to all other DL architectures.	
Effects of image resizers/ interpolators	Faster RCNN ResNet-101	AR with bicubic	0–1.4	10	81.33	Could not contribute to provide better detection results	GitHub
		AR with area	0–0.87	10	91.55	Found as the best interpolator	
		AR with NN	0–0.98	10	86.93	Almost similar performance to the bilinear method	
		FS with bilinear	0–0.92	9.5	85.09	Provided a comparatively lower mAP	
		FS with bicubic	0–1.2	9.5	82.38	Attained a low mAP just like with AR	
		FS with area	0–1.5	9.5	85.68	Area interpolator did not work with fixed shape resizer	
		FS with NN	0–1.4	9.5	82.64	Attained low AP of the weed classes	
Effects of initializers and batch normalization		Tr (std: 0.01); SV (sf: 1.0, nd: true, mode: Fan_avg); RN (std: 0.01)	0–0.87	10	91.55	Very small values of std should not be taken close to zero; the normal distribution with an average of input and output units in the weight tensor should be considered	GitHub
		BN (decay: 0.99, eps: 0.01)	0–0.82	8.5	93.37	An improvement of 1.82% was obtained with BN with a fast training convergence	
Effects of optimizers		SGD with momentum	0–0.87	8.5	93.37	The default optimizer attained a high AP except for the negative clas	GitHub
		Adam	0–0.94	7.75	91.56	Faster convergence with adaptive algorithm	
		RMSProp	0–0.86	7.75	93.44	The best-obtained DL optimizer, slightly improved the mAP without BN	

FS, Fixed-shape resizer; AR, Aspect ratio resizer; NN, Nearest neighbor; Tr, Truncated normal initializer; std, standard deviation; SV, Scaling variance initializer; sf, scaling factor; nd, normal distribution; RN, random normal initializer; BN, Batch normalization; eps, epsilon.

- Subsequently, three interpolation methods were also applied to the aspect ratio resizer. The bicubic method could not contribute to the improvement of mAP or detection results.
- The nearest neighbor's performance was almost similar to the bilinear method. However, the 'area' interpolation method provided significantly better training performance than the Faster RCNN model. It resulted in a higher mAP of 91.55% as presented in **Table 4**.
- Moreover, the average precision of the individual five classes was improved by a significant margin; these classes

include lantana, prickly acacia, parkinsonia, snakeweed, and negative.
- The area interpolation method reduces noise from the images. The final images fed into the network contributed to better feature extraction of the weed classes.
- It was also noticed that the input provided to the Faster RCNN ResNet-101 model with aspect ratio resizer with area interpolation method achieved an improvement in both training and testing performance. This has been shown graphically by various losses that constitute the total training loss. The losses like RPN (Region Proposal Network) and final classifier losses are presented by RPN objectness loss (R_obj_loss), RPN localization loss (R_loc_loss), classification loss (Class_loss), and localization loss (Loc_loss) as shown in **Figure 10**. It is important to consider these losses in the analysis because this research is dedicated to performing the weed detection task, which contains both classification and localization operations.
- From **Figure 10**, it can be concluded that the losses related to the localization and classification tasks were reduced when the area interpolation method was applied. For example, the localization loss (Loc_loss) was settled with a small fluctuation between 0 and 0.52% for the bilinear interpolation method, which got reduced to 0–0.45%. Similarly, the classification loss (Class_loss) was 0–0.35% and reduced to 0–0.3%. Furthermore, RPN losses were improved with the area interpolation method.
- Since both the localization and classification losses were reduced in the region proposal and classifier stages of the network, therefore, total losses were also reduced from 0–1% to 0–0.87%. Hence, a small reduction in the model's losses produced a considerable effect on weed detection results.

In summary, an improvement in various losses along with AP (of individual classes) and mAP indicates that the area interpolation method with the aspect ratio resizer can also be applied to the Faster RCNN model for other relevant datasets. An improvement in detection output proves that the scientific community should focus on further advances in image resizing/interpolation techniques using CNN (Islam et al., 2018).

Weights Optimization

Initialization Techniques

The previous studies have been performed several agricultural tasks by using various advancements in DL including training techniques, augmentation methods, and modifying particular types of hidden convolutional layers of neural networks. In contrast, this research studied the effects of weight optimization methods on the performance of DL models.

An optimized version of the Faster RCNN model is presented by analyzing the effects of weight initializers, batch normalization (BN), and DL optimizers. Firstly, the initialization techniques were studied since they are important for a neural network to prevent vanishing gradients, which is essential to get convergence of the models (Narkhede et al., 2021). It is vital to initialize the weights with neither too large nor too small standard deviation, as both conditions fail the network to learn the features properly. Three weight initializers were studied, including truncated normal, scaling variance, and random normal initializers, according to the names presented in the TensorFlow object detection API. The optimum selection of the weight initializer parameters for the Faster RCNN model and its effects are discussed as follows.

- The truncated normal initializer is the most recommended weight initializing technique for a convolutional neural network due to the use of the ReLU activation function in almost all the networks. The reason is its vanishing gradient solving capability. This initializer is very useful for eliminating dead neurons.
- The truncated normal initializer was used to avoid any value beyond twice the standard deviation. Different values of the standard deviation of the truncated initializer affected the overall model's performance. Initially, the Faster RCNN model was trained with unit standard deviation, but it was an unsuitable value to converge the training. It can be concluded that the higher the value of standard deviation and the closer to 1, the more the training time and the lower mAP would be obtained.
- Therefore, this initializer was used with a standard deviation equal to 0.01 and a zero mean value (Ren et al., 2015). When selecting a lower standard deviation value, it should not be taken very close to zero because the mAP obtained with a standard deviation equal to 0.001 was almost 8% less than the mAP with the selected SD (0.01).
- Then, the random normal initializer was also tested that attained a mAP of 89.07%.
- Later, a scaling variance initializer with a fully convolutional layer (FC) was used. It also contains a few tunable parameters including scaling factor, normal distribution, and three modes of operation depending on input and output units in the weight tensor.
- With the combination of scaling factor 2 without normal distribution and considering only input units in the weight tensor (Fan_In), a considerably good performance in terms of losses and a mAP of 85.47% was observed.
- Furthermore, the scaling factor 1.0 considering the normal distribution of the Fan_Avg mode (which contains an average of the number of input and output units in the weight tensor) was found to be the most appropriate setting to get the best detection results in terms of mAP. **Table 4** presents the mAP (91.55%) having parameter values of the weight initializer described in this section with the aspect ratio resizer by the area interpolation method.

Batch Normalization

The next step was to study the effects of batch normalization (BN) on the performance of the Faster RCNN ResNet-101 model. Training/testing profiles were evaluated in the absence and presence of BN. The following observations were made:

- First, the model was trained with default values of decay and epsilon of 0.99 and 0.001, respectively. The training performance of the Faster RCNN model was improved in terms of a total loss of 0.85%. The iterations were reduced

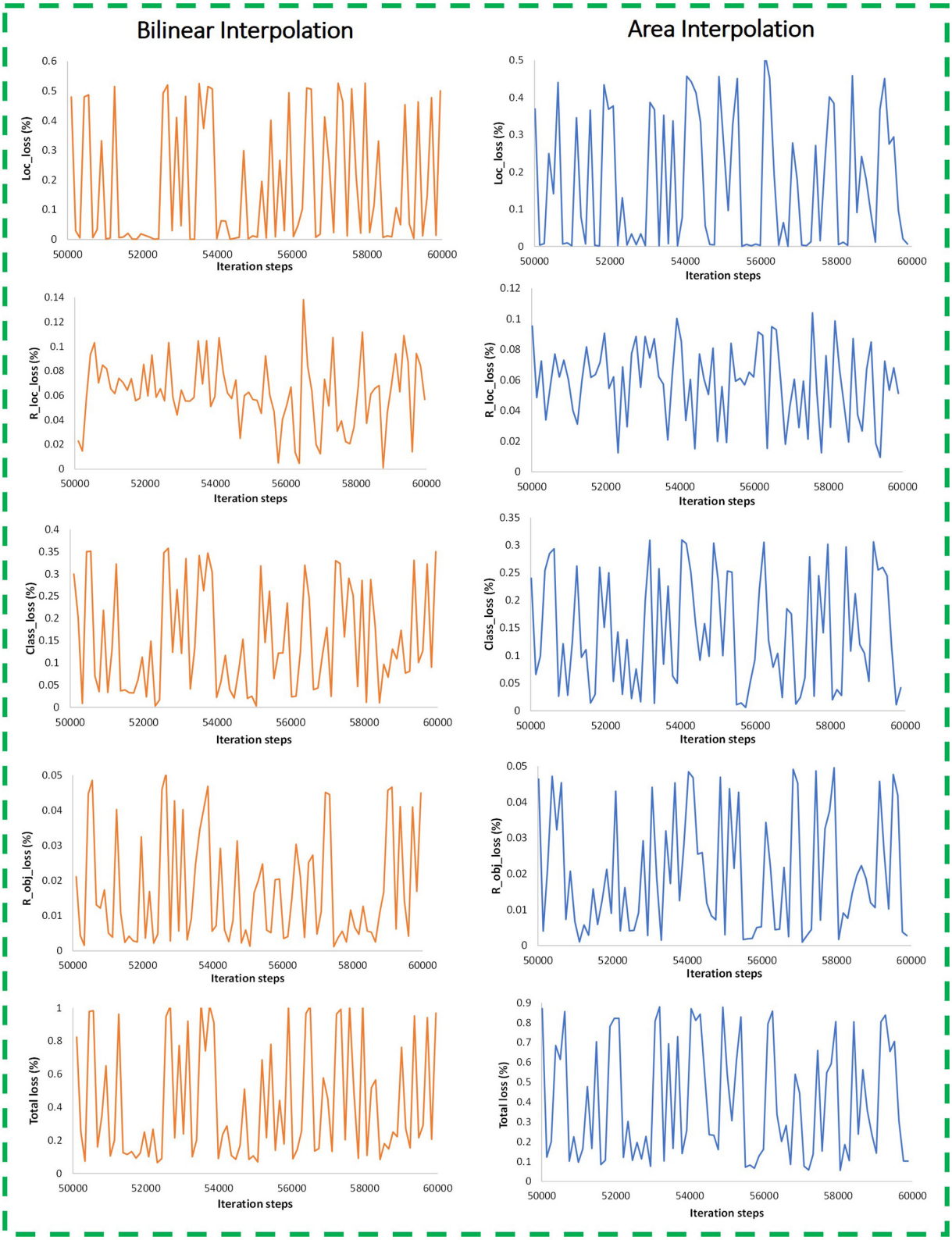


FIGURE 10 | Training plots for bilinear and area interpolation methods with aspect ratio image resizer (iteration steps from 50K and onwards are shown, when the model got training convergence).

to only 47K steps from 50K iterations. It shows the fast convergence of the DL architecture with the application of BN (Santurkar et al., 2018).
- However, the mAP was almost equal as obtained in the previous step of weight initialization with 91.34%.
- Next, the decay and epsilon were tuned, and it was found that a higher value of epsilon improved the overall training/testing performance of the model. For instance, at an epsilon value of 0.01 and default decay value, the mAP was improved with a margin of 1.82% as compared to the former stage of the weight initialization. The total training loss was also reduced to almost 0.82%.
- AP of individual classes was also improved including chinee apple, prickly acacia, parkinsonia, and siam weed with a difference of 22.85, 13.13, 8.99, and 9.39%, respectively. On the other hand, two classes such as lantana and negative (non-weed) attained a lower AP with BN.
- Otherwise, a smaller epsilon (0.0001) degraded the performance of the Faster RCNN model with only 76.83% of mAP. Similarly, the smaller decay could not contribute to improving the mAP and attained 88.46% with 0.5 decay value.

Deep Learning Optimizers

The final step of the study presented in this article is comprised of the optimization of weights by three DL optimizers. Hyperparameters were selected using the random search method (Bergstra and Bengio, 2012) as presented in **Table 2**. The SGD with momentum was the default optimizer for training the Faster RCNN ResNet-101 model. Subsequently, Adam and RMSProp were used to optimize the weights of the model. A class-wise performance of the DL optimizers is presented in **Figure 11**. The effects of all three optimizers are discussed below.

- First, the DL optimizers were analyzed with and without BN. In the presence of BN, the SGD with momentum and Adam optimizers improved the performance with 1.82 and 1.05% in mAP, respectively, compared to the results obtained without BN. However, RMSProp got a reduction of 0.5% in mAP with BN.
- The RMSProp attained the highest mAP of 93.44% in the absence of BN. However, the SGD optimizer attained a comparable mAP of 93.37% with BN. Due to a slight difference of mAP by RMSProp (without BN), it was selected as the best DL optimizer. Moreover, RMSProp attained a much higher AP of classes including lantana and negative (non-weed), as shown in **Figure 11**.
- In terms of the number of training steps, the Faster RCNN model required a lesser number of steps with BN. For example, the SGD optimizer required around 47K steps to obtain the model convergence. However, adaptive optimizers including Adam and RMSProp required around 44K steps and showed a faster training convergence (Zhou et al., 2020).
- Few classes attained the same/high average precision with the three DL optimizers, such as parthenium and rubber vine. It suggests that these classes of weeds should not be addressed in future studies.
- It is also noticed that all classes achieved an AP of more than 90% except for chinee apple, when the model was trained with RMSProp. Therefore, it can be concluded that the features of this particular class of weed were not extracted through RMSProp.
- The effectiveness of the fine-tuned adaptive method like RMSProp has been shown from these experiments. Furthermore, the significance of the random search method (Bergstra and Bengio, 2012) has also been evidenced for not only the learning rate but also for the other hyperparameters, including decay factor, momentum, and epsilon (presented in **Table 2**).
- The stratified k-fold cross-validation technique was used to further confirm the final mAP of 93.44%. In this regard, the dataset images were redistributed in five folds (fold1-fold5). The initial distribution of the dataset was considered the first fold (fold1). The optimized Faster RCNN ResNet-101 model was retrained with the rest of the four-folds. A small difference in mAP was observed from 0.14 to 0.46%, and attained 93.30, 93.84, 93.71, and 92.98% using fold2, fold3, fold4, and fold5, respectively.

To conclude, an overall summary of all the experiments performed for this research is presented in **Table 4**, including the model assessment on training/testing datasets and relevant comments indicating the significance of each step taken.

CONCLUSIONS AND FUTURE RECOMMENDATIONS

This article presents a deep learning-based approach consisting of five steps for the detection of weeds. First, an open-source dataset called DeepWeeds was selected due to its dynamic nature, which considered various practical aspects of an agricultural field. Next, the performance of various single-stage and two-stage neural networks was evaluated. After an in-depth analysis of the DL architectures, Faster RCNN trained with the ResNet-101 feature extractor model achieved the highest mAP of 87.64%. Later, several attempts were made to improve the class-wise average precision of the best-obtained DL model. Formerly, the effects of image resizing techniques and image interpolation methods were studied. The aspect ratio resizer with the area interpolation method attained the highest mAP of 91.55% which was 3.91% better than the default settings. Furthermore, the training performance of the Faster RCNN model was also enhanced in terms of various classification and localization losses. Next, weight-optimization techniques were thoroughly studied. In this regard, the effects of weight initializers, including truncated normal, scaling variance, and random normal, were evaluated. The performance of the model was also analyzed with batch normalization and an enhancement of 1.82% in mAP was observed. Finally, the adaptive DL optimizers including Adam and RMSProp were used to retrain the Faster RCNN model. The optimal selection of hyperparameters of the RMSProp optimizer slightly improved the mAP by 93.44%. Hence, an improvement of 5.8% mAP in the best-obtained DL architecture was achieved

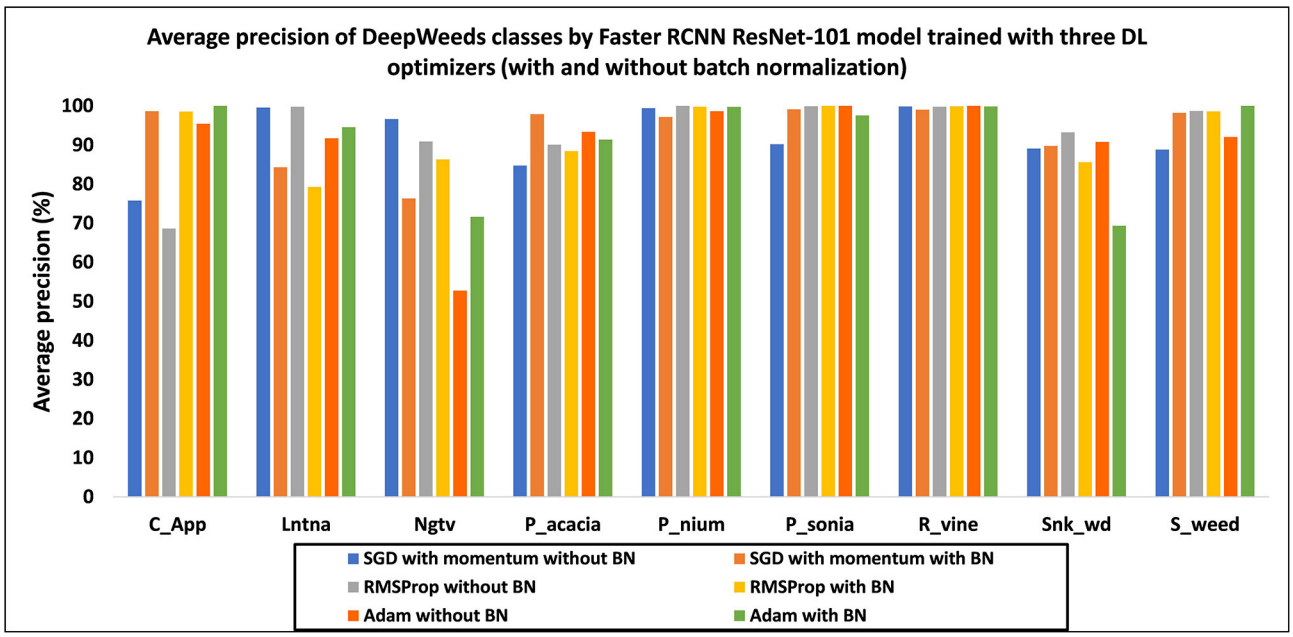


FIGURE 11 | Average precision of each class by the Faster RCNN ResNet-101 model trained with three DL optimizers in the absence and presence of batch normalization.

as compared to the default settings and it proved the effectiveness of the weed detection pipeline presented in this article.

The methodology presented in this research would be a considerable step toward precision agriculture due to a significant improvement in a complex agricultural task such as weed identification. Furthermore, this research also provides various future directions to further enhance the agricultural field of research by DL:

- The chinee apple class attained the lowest average precision with the optimized settings of the Faster RCNN model. Therefore, future research could include the modification of the Faster RCNN model to extract the unique feature of the chinee apple and maintain the average precision of other weed classes. For example, the Faster RCNN ResNet-101 contains a region proposal network and a classification model (ResNet-101). The ResNet-101 has several hidden layers to extract the distinct features of the objects. Therefore, an in-depth analysis of various hyperparameters of ResNet-101 could be performed including the number of hidden layers, filter size, number of strides, and using the latest advancement in activation functions.
- This research could be useful for other agricultural applications, including detection of plant diseases, classification of agricultural land cover, recognition of fruits, etc. After analyzing the performance of the single-stage and two-stage DL object detectors, the proposed DL-based study can be treated as an intermediate step before proposing any modification in the DL architecture.
- The weights obtained by the final optimized Faster RCNN model can be reused as transfer learning to other weed-related datasets.
- This research was dedicated to improving the final mAP of the Faster RCNN ResNet-101 model. Future research could also attempt to analyze/reduce the computation/training time and real-time detection of weeds.
- Other advanced DL optimizers can be used for upcoming studies such as Ranger optimizer.

DATA AVAILABILITY STATEMENT

The annotation files of the DeepWeeds dataset, configuration files for all models, inference graph of the final model, and dataset for five-fold cross-validation method are made publicly available in a GitHub repository https://github.com/kmarif/DL-Weed-Identification.

AUTHOR CONTRIBUTIONS

MHS and KMA designed the research. MHS proposed the methodology, performed the experiments, wrote the original draft, and prepared the revision. KKV curated the data and contributed to getting the results. KMA reviewed and edited the manuscript. KMA and JP obtained funding and supervised this research. All authors contributed to the article and approved the submitted version.

FUNDING

This research was funded by the Ministry of Business, Innovation, and Employment (MBIE), New Zealand, Science for Technological Innovation (SfTI) National Science Challenge.

REFERENCES

Ahmad, A., Saraswat, D., Aggarwal, V., Etienne, A., and Hancock, B. (2021). Performance of deep learning models for classifying and detecting common weeds in corn and soybean production systems. *Comput. Electron. Agric.* 184, 106081. doi: 10.1016/j.compag.2021.106081

Bergstra, J., and Bengio, Y. (2012). Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* 13.

Bisen, D. (2021). Deep convolutional neural network based plant species recognition through features of leaf. *Multimed. Tools Appl.* 80, 6443–6456. doi: 10.1007/s11042-020-10038-w

Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M. (2020). Yolov4: optimal speed and accuracy of object detection. *arXiv.* 2004, 10934. doi: 10.48550/arXiv.2004.10934

Caldera, U., and Breyer, C. (2019). Assessing the potential for renewable energy powered desalination for the global irrigation sector. *Sci. Total Environ.* 694, 133598. doi: 10.1016/j.scitotenv.2019.133598

Chandel, N. S., Chakraborty, S. K., Rajwade, Y. A., Dubey, K., Tiwari, M. K., and Jat, D. (2021). Identifying crop water stress using deep learning models. *Neural. Comput. Appl.* 33, 5353–5367. doi: 10.1007/s00521-020-05325-4

Dai, J., Li, Y., He, K., and Sun, J. (2016). "R-fcn: Object detection via region-based fully convolutional networks", in *Advances in neural information processing systems* (Spain: Curran Associates Inc.), *p*. 379–387.

De Camargo, T., Schirrmann, M., Landwehr, N., Dammer, K.-H., and Pflanz, M. (2021). Optimized deep learning model as a basis for fast UAV mapping of weed species in winter wheat crops. *Remote Sens.* 13, 1704. doi: 10.3390/rs13091704

Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., and Tian, Q. (2019). "Centernet: Keypoint triplets for object detection", in *Proceedings of the IEEE/CVF International Conference on Computer Vision*. p. 6569–6578. doi: 10.1109/ICCV.2019.00667

Espejo-Garcia, B., Mylonas, N., Athanasakos, L., Fountas, S., and Vasilakoglou, I. (2020). Towards weeds identification assistance through transfer learning. *Comput. Electron. Agric.* 171, 105306. doi: 10.1016/j.compag.2020.105306

Fu, L., Feng, Y., Wu, J., Liu, Z., Gao, F., Majeed, Y., et al. (2021). Fast and accurate detection of kiwifruit in orchard using improved YOLOv3-tiny model. *Precis. Agric.* 22, 754–776. doi: 10.1007/s11119-020-09754-y

Gai, R., Chen, N., and Yuan, H. (2021). A detection algorithm for cherry fruits based on the improved YOLO-v4 model. *Neural. Comput. Appl.* 1–12. doi: 10.1007/s00521-021-06029-z

Gao, J., French, A. P., Pound, M. P., He, Y., Pridmore, T. P., and Pieters, J. G. (2020). Deep convolutional neural networks for image-based Convolvulus sepium detection in sugar beet fields. *Plant Methods.* 16, 1–12. doi: 10.1186/s13007-020-00570-z

Hasan, A. M., Sohel, F., Diepeveen, D., Laga, H., and Jones, M. G. (2021). A survey of deep learning techniques for weed detection from images. *Comput. Electron. Agric.* 184, 106067. doi: 10.1016/j.compag.2021.106067

He, H., and Ma, Y. (2013). *Imbalanced Learning: Foundations*, Algorithms, and Applications. doi: 10.1002/9781118646106

He, K., Zhang, X., Ren, S., and Sun, J. (2015). "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification", *in Proceedings of the IEEE International Conference on Computer Vision*. p. 1026–1034. doi: 10.1109/ICCV.2015.123

Hinton, G., Srivastava, N., and Swersky, K. (2012). Neural networks for machine learning. *Coursera, Video Lectures.* 264, 2146–2153.

Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., et al. (2016). "Speed/accuracy trade-offs for modern convolutional object detectors", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. p. 7310–7311. doi: 10.1109/CVPR.2017.351

Ioffe, S., and Szegedy, C. (2015). "Batch normalization: accelerating deep network training by reducing internal covariate shift", in *International conference on machine learning*. PMLR. p. 448–456.

Islam, M. T., Rahman, S. M., Ahmad, M. O., and Swamy, M. (2018). Mixed Gaussian-impulse noise reduction from images using convolutional neural network. *Signal Process. Image Commun.* 68, 26–41. doi: 10.1016/j.image.2018.06.016

Jiang, H., Zhang, C., Qiao, Y., Zhang, Z., Zhang, W., and Song, C. (2020). CNN feature based graph convolutional network for weed and crop recognition in smart farming. *Comput. Electron. Agric.* 174, 105450. doi: 10.1016/j.compag.2020.105450

Jin, X., Che, J., and Chen, Y. (2021). Weed identification using deep learning and image processing in vegetable plantation. *IEEE Access.* 9, 10940–10950. doi: 10.1109/ACCESS.2021.3050296

Khan, S., Tufail, M., Khan, M. T., Khan, Z. A., and Anwar, S. (2021). Deep learning-based identification system of weeds and crops in strawberry and pea fields for a precision agriculture sprayer. *Precis. Agri.* 1–17. doi: 10.1007/s11119-021-09808-9

Kingma, D. P., and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv.* 1412.6980.

Knoll, F. J., Czymmek, V., Harders, L. O., and Hussmann, S. (2019). Real-time classification of weeds in organic carrot production using deep learning algorithms. *Comput. Electron. Agric.* 167, 105097. doi: 10.1016/j.compag.2019.105097

Kounalakis, T., Triantafyllidis, G. A., and Nalpantidis, L. (2019). Deep learning-based visual recognition of rumex for robotic precision farming. *Comput. Electron. Agric.* 165, 104973. doi: 10.1016/j.compag.2019.104973

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* 25, 1097–1105.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017). "Focal loss for dense object detection", in *Proceedings of the IEEE international conference on computer vision*. p. 2980–2988. doi: 10.1109/ICCV.2017.324

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., et al. (2015). "Ssd: Single shot multibox detector", in *European conference on computer vision*. Amsterdam: Springer. *p*. 21–37. doi: 10.1007/978-3-319-46448-0_2

Lottes, P., Behley, J., Chebrolu, N., Milioto, A., and Stachniss, C. (2020). Robust joint stem detection and crop-weed classification using image sequences for plant-specific treatment in precision farming. *J. Field Robot.* 37, 20–34. doi: 10.1002/rob.21901

Malik, A., Sikka, G., and Verma, H. K. (2017). An image interpolation based reversible data hiding scheme using pixel value adjusting feature. *Multimed. Tools Appl.* 76, 13025–13046. doi: 10.1007/s11042-016-3707-5

Masters, D., and Luschi, C. (2018). Revisiting small batch training for deep neural networks. *arXiv.* 1804, 07612. doi: 10.48550/arXiv.1804.07612

Narkhede, M. V., Bartakke, P. P., and Sutaone, M. S. (2021). A review on weight initialization strategies for neural networks. *Artif. Intell. Rev.* 1–32. doi: 10.1007/s10462-021-10033-z

Olsen, A., Konovalov, D. A., Philippa, B., Ridd, P., Wood, J. C., Johns, J., et al. (2019). DeepWeeds: A multiclass weed species image dataset for deep learning. *Scientific Rep.* 9, 1–12. doi: 10.1038/s41598-018-38343-3

Osorio, K., Puerto, A., Pedraza, C., Jamaica, D., and Rodríguez, L. (2020). A deep learning approach for weed detection in lettuce crops using multispectral images. *Agri Eng.* 2, 471–488. doi: 10.3390/agriengineering2030032

Priyadharshini, R. A., Arivazhagan, S., Arun, M., and Mirnalini, A. (2019). Maize leaf disease classification using deep convolutional neural networks. *Neural. Comput. Appl.* 31, 8887–8895. doi: 10.1007/s00521-019-04228-3

Quiroz, I. A., and Alférez, G. H. (2020). Image recognition of Legacy blueberries in a Chilean smart farm through deep learning. *Comput. Electron. Agric.* 168, 105044. doi: 10.1016/j.compag.2019.105044

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* 28, 91–99.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv.* 1609.04747. doi: 10.48550/arXiv.1609.04747

Saleem, M. H., Khanchi, S., Potgieter, J., and Arif, K. M. (2020a). Image-based plant disease identification by deep learning meta-architectures. *Plants.* 9, 1451. doi: 10.3390/plants9111451

Saleem, M. H., Potgieter, J., and Arif, K. M. (2019). Plant disease detection and classification by deep learning. *Plants.* 8, 468. doi: 10.3390/plants8110468

Saleem, M. H., Potgieter, J., and Arif, K. M. (2020b). Plant disease classification: a comparative evaluation of convolutional neural networks and deep learning optimizers. *Plants.* 9, 1319. doi: 10.3390/plants9101319

Saleem, M. H., Potgieter, J., and Arif, K. M. (2021). Automation in agriculture by machine and deep learning techniques: a review of recent developments. *Precis. Agri.* 1–39. doi: 10.1007/s11119-021-09806-x

Santurkar, S., Tsipras, D., Ilyas, A., and Madry, A. (2018). How does batch normalization help optimization? *Adv. Neural Inf. Process. Syst.* 31. doi: 10.48550/arXiv.1805.11604

Tan, M., Pang, R., and Le, Q. V. (2019). "Efficientdet: Scalable and efficient object detection", in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (Seattle: IEEE), 10781–10790. doi: 10.1109/CVPR42600.2020.01079

Uguz, S., and Uysal, N. (2021). Classification of olive leaf diseases using deep convolutional neural networks. *Neural. Comput. Appl.* 33, 4133–4149. doi: 10.1007/s00521-020-05235-5

Veeranampalayam Sivakumar, A. N., Li, J., Scott, S., Psota, E., J., Jhala, A., et al. (2020). Comparison of object detection and patch-based classification deep learning models on mid-to late-season weed detection in UAV imagery. *Remote Sens.* 12, 2136. doi: 10.3390/rs12132136

Yu, J., Schumann, A. W., Cao, Z., Sharpe, S. M., and Boyd, N. S. (2019a). Weed detection in perennial ryegrass with deep learning convolutional neural network. *Front. Plant Sci.* 10, 1422. doi: 10.3389/fpls.2019.01422

Yu, J., Sharpe, S. M., Schumann, A. W., and Boyd, N. S. (2019b). Deep learning for image-based weed detection in turfgrass. *Eur. J. Agron.* 104, 78–84. doi: 10.1016/j.eja.2019.01.004

Yu, J., Sharpe, S. M., Schumann, A. W., and Boyd, N. S. (2019c). Detection of broadleaf weeds growing in turfgrass with convolutional neural networks. *Pest Manag. Sci.* 75, 2211–2218. doi: 10.1002/ps.5349

Zhou, P., Feng, J., Ma, C., Xiong, C., and Hoi, S. C. H. (2020). Towards theoretically understanding why sgd generalizes better than adam in deep learning. *Adv. Neural Inf. Process. Syst.* 33, 21285–21296.

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Chapter 6 Weed Detection by Faster RCNN Model: An Enhanced Anchor Box Approach

This chapter contains content from the following article.

M. H. Saleem, J. Potgieter, and K. M. Arif, " Weed detection by Faster RCNN model: An enhanced anchor box approach," vol. 12, no. 7, p. 1580, *Agronomy*, 2022. https://doi.org/10.3390/agronomy12071580

agronomy

MDPI

Article

Weed Detection by Faster RCNN Model: An Enhanced Anchor Box Approach

Muhammad Hammad Saleem [1], Johan Potgieter [2] and Khalid Mahmood Arif [1,*]

[1] Department of Mechanical and Electrical Engineering, School of Food and Advanced Technology, Massey University, Auckland 0632, New Zealand; H.Saleem@massey.ac.nz
[2] Massey AgriFood Digital Lab, Massey University, Palmerston North 4472, New Zealand; J.Potgieter@massey.ac.nz
* Correspondence: K.Arif@massey.ac.nz

Abstract: To apply weed control treatments effectively, the weeds must be accurately detected. Deep learning (DL) has been quite successful in performing the weed identification task. However, various aspects of the DL have not been explored in previous studies. This research aimed to achieve a high average precision (AP) of eight classes of weeds and a negative (non-weed) class, using the DeepWeeds dataset. In this regard, a DL-based two-step methodology has been proposed. This article is the second stage of the research, while the first stage has already been published. The former phase presented a weed detection pipeline and consisted of the evaluation of various neural networks, image resizers, and weight optimization techniques. Although a significant improvement in the mean average precision (mAP) was attained. However, the Chinee apple weed did not reach a high average precision. This result provided a solid ground for the next stage of the study. Hence, this paper presents an in-depth analysis of the Faster Region-based Convolutional Neural Network (RCNN) with ResNet-101, the best-obtained model in the past step. The architectural details of the Faster RCNN model have been thoroughly studied to investigate each class of weeds. It was empirically found that the generation of anchor boxes affects the training and testing performance of the Faster RCNN model. An enhancement to the anchor box scales and aspect ratios has been attempted by various combinations. The final results, with the addition of 64 × 64 scale size, and aspect ratio of 1:3 and 3:1, produced the best classification and localization of all classes of weeds and a negative class. An enhancement of 24.95% AP was obtained in Chinee apple weed. Furthermore, the mAP was improved by 2.58%. The robustness of the approach has been shown by the stratified k-fold cross-validation technique and testing on an external dataset.

Keywords: deep learning; convolutional neural network; weed detection; optimization algorithms; transfer learning

Citation: Saleem, M.H.; Potgieter, J.; Arif, K.M. Weed Detection by Faster RCNN Model: An Enhanced Anchor Box Approach. *Agronomy* **2022**, *12*, 1580. https://doi.org/10.3390/agronomy12071580

Academic Editor: Wen-Hao Su

Received: 22 May 2022
Accepted: 27 June 2022
Published: 29 June 2022

1. Introduction

Conventional weed control methods are generally cost-ineffective and produce adverse effects on the environment [1]. Hence, there is a need for an automatic weed control system that can reduce human and machinery efforts. Before deploying a weed management system, a precise identification/detection of weeds is a mandatory task. In this regard, computer vision plays a vital role in conjunction with artificial intelligence. Deep learning (DL), a class of machine learning (ML), has been a promising tool for performing various real-life object detection tasks, including agricultural operations, such as identification of plant diseases [2], agricultural land cover classification [3], fruit recognition [4], plant recognition [5], and many others [6].

Recently, weed recognition has also been focused on by the research community. Several DL-based methods have been introduced to detect various weed classes. A recent review paper presents a summary of the characteristics along with the advantages and

disadvantages of traditional ML-based techniques, deep learning algorithms, and available datasets and weeding equipment [7]. At the beginning of research into DL-based weed identification, most of the studies were focused on the weed classification task by DL-based feature extractors. For instance, one study presents a combined convolutional neural network (CNN) with a support vector machine (SVM) for the recognition of weeds [8]. Similarly, other research has proposed a graph-based CNN with ResNet-101 for weed classification [9]. Hu et al. [10] also presented a graph-based DL model to classify weeds. Then, the research community started to focus on both classification and localization tasks using DL-based object detection methods. For example, a research article used two DL models including EfficientDet and a version of You Only Look Once (YOLO-v5) to detect monocot/dicot weeds [11]. For that research, in-field images were collected that showed the usefulness of the state-of-the-art deep learning models for real/complex environments. On the other hand, loss plots should also be presented to show the training performance of the models. Furthermore, some of the older DL architectures could also be tested to show the significance of the selected model. Various DL classification models such as GoogLeNet, VGG, and DetectNet were used to detect broadleaf weeds [12]. The research evaluated various feature extractors, but training profiles that could be useful to understand the training performance of the DL models were not presented. A research showed the importance of the YOLO-v3 architecture for weed identification, and the performance of two commonly used DL frameworks was evaluated, including TensorFlow and Keras [13]. The trained model successfully detects weeds, but the selection of YOLO-v3 was not justified clearly. Hence, a comparative analysis with other object detection methods could further support the selection of the YOLO-v3 model. A recent article addressed an important task of weed identification in the seedling growth stage using DL architectures [14]. That study not only addressed an important agricultural problem but also evaluated different DL classification and detection models along with various considerations of image sizes. However, training performance should also be analyzed. Gao et al. [15] proposed a reduced model of YOLO for the recognition of weeds and proved the usefulness of the proposed approach by comparing it with former versions of YOLO. That research provided inference time for the proposed methods, which is one of the main practical aspects of the implementation of such research. The CenterNet model was used to detect weeds in [16].

The research community also evaluated transfer learning for weed recognition. The DL models were used in research for classification purposes along with conventional/traditional ML techniques to detect several classes of weeds [17]. A study presented the performance of 35 DL models through transfer learning for the identification of weeds in the real environment [18]. Suh et al. [19] presented a transfer learning-based approach to classifying weeds using pre-trained weights on ImageNet. One of the key features of that research was the evaluation of DL models on the images collected at different periods. It shows the strength of DL for weed detection. Although the transfer learning methods produced significant results, the main research gap was that the proposed approaches have not been tested on an external dataset with the same weed/crop classes. This would show the robustness of the methodologies in different field environments. Few of the studies analyzed the significance of DL model for Unmanned Aerial Vehicles (UAVs). A study investigated images of UAVs to recognize weeds using a ResNet-based model [20]. Research has been conducted on the identification of weeds using the ResNet model [21]. Ukaegbu et al. [22] used a CNN model on a quadcopter to detect broadleaf and grassweeds and to evaluate herbicide spraying. A UAV was used to detect weeds using Faster RCNN and SSD models [23]. An article presented an improved version of the Faster RCNN model for the image collected by UAV for weed identification [24]. The performance of the proposed method was compared with other DL and ML methods. However, the specifications of the anchor boxes were not analyzed in detail, which is a major research gap. Few robotic platforms have also been proposed for weed detection by DL. For example, the YOLO-v3 model with a DarkNet-53 feature extractor was trained and tested on a mobile platform to identify crops and weeds in a practical field, and a chemical spray was applied

to the detected weeds [25]. Kounalakis et al. [26] designed a prototype robotic system for weed detection using a CNN model. Furthermore, few studies have been conducted to perform weed segmentation [27–29], but it is not the focus of this research.

From the literature presented earlier, it can be summarized that most of the previous studies relied on either proposing the backbone models/feature extractors or leveraging the state-of-the-art DL object detection methods for weed identification purposes. However, to the best of our knowledge, none of the previous approaches has provided a systematic way to analyze the robustness of deep learning by exploiting various aspects of image resizing and optimization methods for the recognition of weeds. Moreover, a detailed analysis of a well-known object detector, Faster Region-based Convolutional Neural Network (RCNN), has not been performed at its architectural level to successfully detect weeds. Therefore, a DL-based approach has been proposed for weed identification, divided into two stages. The first stage has already been published in another journal. It presented a weed detection framework and obtained the best DL architecture along with the selection of the most optimum image resizer, interpolator, weights initializer, and DL optimizer (in the absence and presence of batch normalization) [30]. However, this paper is dedicated to the second stage of the research. The proposed method is based on our observations during the initial stage of the study.

It was found during the former phase of the study that all weed classes attained a high (more than 90%) average precision (AP), except for the Chinee Apple class from the selected dataset called DeepWeeds [31]. The Faster RCNN ResNet-101 model was found to be the most suitable method trained with the RMSProp optimizer and aspect ratio resizer technique with area interpolation. A high AP of seven weed classes showed that the ResNet-101 performed well in extracting the unique features of the weed classes. However, the distinct features of the Chinee apple weed were not well extracted. Few of the test images belonging to the Chinee apple were detected with a high confidence score. However, most of the images of the Chinee apple were undetected by the trained model. Hence, there was a need to further investigate the Faster RCNN model to detect the remaining images of the Chinee apple, along with maintaining the high AP of the other weed classes (and a negative/non-weed class).

The main contributions of this research are:

1. Obtained a 24.95% higher AP for Chinee apple and maintained the performance of all other weed classes and a negative class with AP > 89%;
2. Investigated another robust way to improve the weed detection task by deep learning;
3. Evaluated the significance of various anchor box scales and aspect ratios for weed identification that can be replicated for other agricultural applications;
4. Achieved an improvement of 2.58% in mean average precision compared to the former phase of the research;
5. Shown the robustness of the approach by the stratified k-fold cross-validation technique and testing on an externally generated dataset;
6. Made all data including the final weights of the optimized Faster RCNN model publicly available to reuse as a transfer learning for other weeds-related datasets.

The rest of the paper is organized as follows: Section 2 describes the dataset, training, specifications, and steps of the enhanced anchor boxes. Section 3 presents the summary of the first phase of the research and results and discussion of the proposed approach by graphical/training plots, and detection outcomes. Section 4 concludes the research along with future directions.

2. Materials and Methods

This section elaborates on the selected dataset, specifications related to the deep learning setup, and methodology for the enhancement in the anchor boxes.

The overall flow of both stages of the research is presented in Figure 1.

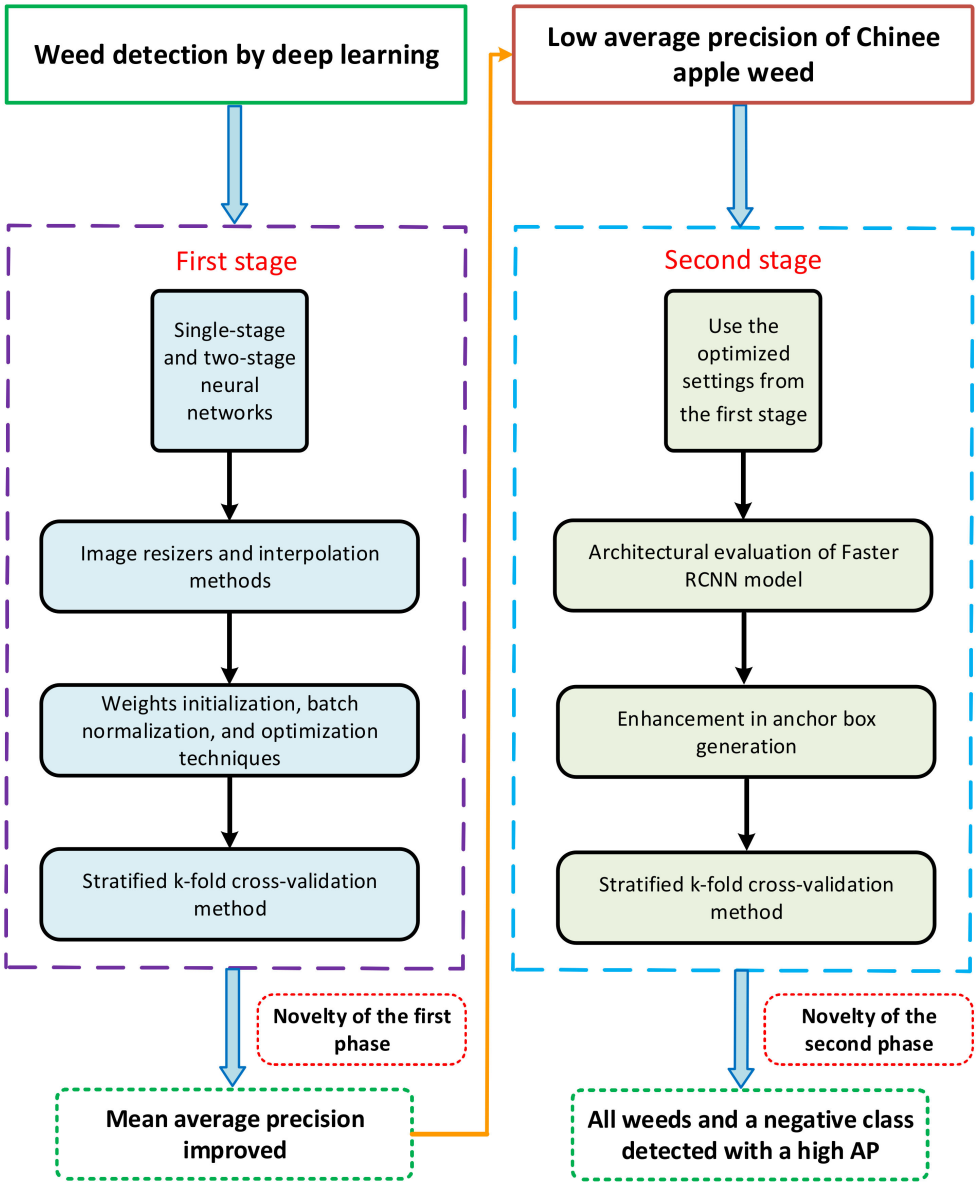


Figure 1. The overall methodology of the two-step DL-based weed identification.

2.1. Dataset Specifications

The DeepWeeds dataset [31] was used throughout this research. This dataset contains images of eight weed classes including a negative/non-weed class, collected in Northern Australia. The reasons for the dataset selection were the diverse nature and consideration of various properties of the real field environment, including actual background, inconsistent lighting, occlusion, etc. Hence, the high detection precision of each class in such a dynamic dataset would show the effectiveness/robustness of the deep learning-based method for weed identification purpose. The dataset contains 17,509 images; it was divided into three sub-datasets for training (70%), validation (20%), and testing (10%). The class names were shortened to visualize the detected results more clearly, such as Chinee apple was annotated as C_App, Lantana was annotated as Lntna, Prickly acacia was annotated by P_acacia, Parthenium with P_nium, Rubber vine with R_vine, Siam weed with S_weed, Parkinsonia with P_sonia, Snakeweed with Snk_wd, and Negative with Ngtv. For the annotations of the dataset images, the XML files were obtained using an open-source tool named LabelImg. These XML files were later converted into CSV and then TF records [32].

2.2. Deep Learning Specifications

From the first step of the research [30], the Faster RCNN model was found to be the best model. It was trained through TensorFlow Object Detection API 1. The transfer learning technique was applied using the weights of the Common Objects in Context (COCO) dataset [33]. Furthermore, all training was conducted using NVIDIA GeForce GTX 1080 Ti GPU card. The most suitable batch size for the Faster RCNN model was equal to 2 [30]. Moreover, the hyperparameters used in the DL optimization method (learning rate, epsilon, momentum, and decay) were selected using the random search technique [34]. According to the methodology presented for the first phase of the research, different DL optimizers were used including Stochastic Gradient Descent (SGD) with momentum, Root Mean Square Propagation (RMSProp), and Adaptive Moment Estimation (Adam). RMSProp attained the best results in terms of mean average precision with the hyperparameters: learning rate = 3×10^{-4}, momentum = 0.9, rho = 0.9, and epsilon = 1.0.

The performance of the DL model was evaluated in terms of average precision (AP) for each class obtained by the 11-point interpolation method [35] and defined as AP at unique recall levels. This method first evaluates the precision at different recall levels and then an interpolated precision is evaluated by taking the maximum/highest precision for a specific recall level. Finally, the average of the AP values for each class is calculated and the mean average precision (mAP) is evaluated. More details can be seen in [35].

Validation of the final results was performed using the stratified k-fold cross-validation method. This technique was adopted due to the class imbalance problem of the DeepWeeds dataset as the negative class has a considerably higher number of images than all weed classes. This method maintains the class distribution in each fold [30]. Furthermore, an external dataset has been generated by random internet search to test the approach in different environments.

2.3. Selection of the Faster RCNN ResNet-101

The first phase of the research evaluated various DL models to select the best-suited architecture for weed detection. In this regard, models including YOLO-v4, Single-Shot MultiBox Detector (SSD) with Inception-v2, MobileNet, and ResNet-50, EfficientDet, CenterNet ResNet-50, Region-based Fully Convolutional Networks (RFCN) ResNet-101, Faster Region-based Convolutional Neural Networks (RCNN) with feature extractors such as Inception-v2, ResNet-50, and ResNet-101, were trained and tested on DeepWeeds dataset [30]. The Faster RCNN ResNet-101 attained superior performance due to its highest mean average precision of 87.64% [30]. Therefore, performance optimization was attempted for the Faster RCNN model in the previous stage and further enhancement has been presented in this article.

2.4. Methodology of the Enhanced Anchor Box Approach

The comprehensive analysis presented in the first stage of the research provided a solid ground for investigating the performance of Faster RCNN in more detail. Careful observation of the detected images of the Chinee apple led to the conclusion that some of the test images were identified with a high confidence score, but most of them remain undetected. To cope with this problem, the main architecture of the Faster RCNN has been thoroughly investigated. The generation of the anchor boxes is one of the main characteristics of the Faster RCNN model. Therefore, an in-depth analysis of anchor box scales and aspect ratios has been performed to improve the AP of the Chinee apple. Furthermore, the AP of all other classes was aimed to be maintained.

2.4.1. Major Novelty of the Original Faster RCNN Model

The weakness of the former version of the Faster RCNN model was its slow speed due to the use of the selective search (SS) method to generate the region proposals [36]. It was addressed through the Region Proposal Network (RPN) in the Faster RCNN model. This version of RCNN takes input images to a feature extractor through a convolutional

Agronomy **2022**, *12*, 1580

neural network model (CNN). Then, the feature map from the convolutional layer is fed to the RPN that generates region proposals in the form of a sliding window. Then, a similar structure comes to its previous model (Fast RCNN), including Region of Interest (ROI) pooling, a classifier, and a bounding box regressor. The main concept and characteristics of the original Faster RCNN are summarized as follows:

- A new network named Region Proposal Network (RPN) was introduced that generates proposals with different scales and aspect ratios.
- In contrast to the Fast RCNN model, the region proposals can be modified according to the specific application.
- The Faster RCNN model combines the RPN and Fast RCNN model; the same convolutional layers are shared between the parts of the model. Hence, no additional time is required to generate the proposals.
- The concept of an anchor box was developed, which is a reference box of specific size and aspect ratio. While training RPN, the training images were passed through the sliding windows using various anchor boxes. The dimensions of the anchor boxes were specified in terms of scale size and aspect ratio and placed in the center of the sliding window. The candidate boxes are then obtained using anchor boxes that operate for the regression. Hence, the anchor box contributes to identifying and localizing the objects with varying dimensions/coordinates.
- The generation of the anchor box was one of the key elements behind the success of the Faster RCNN model. These boxes help solve multiclass classification problems, detect objects of variable size in the dataset, and identify overlapping objects. These reference boxes are placed at various points in the image.

2.4.2. Steps to Obtain Enhanced Anchor Boxes

As discussed in the previous subsection, the generation of several anchor boxes plays a vital role in the detection of target objects. Various aspect ratios and scales together generate anchor boxes, and these boxes make a sliding window that passes through the images.

While annotating the training images from the DeepWeeds dataset, it was empirically observed that the weed classes vary in terms of their bounding box coordinates. Therefore, an enhancement in the anchor box has been attempted through this research to provide a better weed detection outcome. This idea led to a range of new experiments with Faster RCNN to successfully detect and classify all classes of weeds, including the Chinee apple. A detailed explanation of the enhanced anchor box approach is shown in Figure 2.

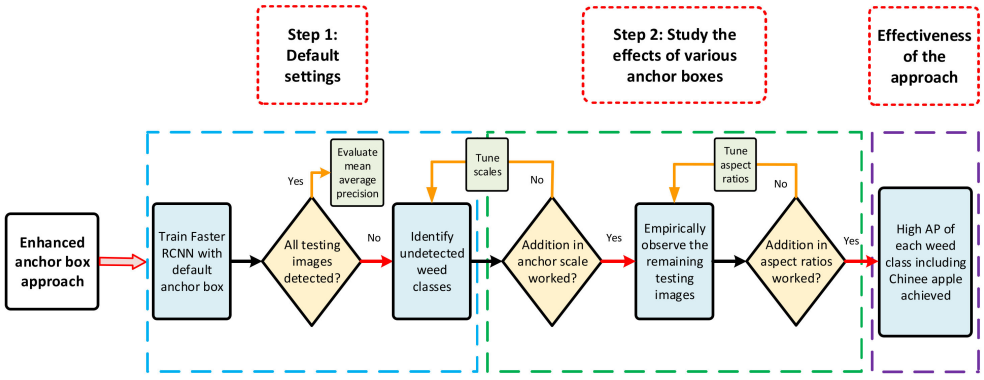


Figure 2. An explanation of the enhanced anchor box approach.

The overall flow of this approach is explained as under:

- First, Faster RCNN was trained on the DeepWeeds dataset with the default scale size and aspect ratio of the anchor boxes according to the original Faster RCNN model. In

this regard, anchor boxes with 128 × 128, 256 × 256, and 512 × 512 scales combined with aspect ratios of 1:1, 2:1, 1:2, and only 1:1 were considered.

- Then, an empirical observation was made to observe whether all testing images are detected or not. In the case of finding undetected images, an enhancement was attempted in the anchor boxes according to the features/characteristics of the weed classes.
- Primarily, the anchor boxes were enhanced by adding one scale. If the insertion of a scale fits the anchor boxes to detect the weed classes, those scales were fixed for the rest of the analysis. Otherwise, more scales were added to obtain the optimum combination. The addition of the scales was performed in the form of twice/half of the previous scale size. For example, the default sizes were 128 × 128, 256 × 256, and 512 × 512; then, a 64 × 64 scale size was inserted. The effects of adding a scale were evaluated in terms of training loss, mAP, and AP of individual classes.
- If the mAP was improved significantly, again, the remaining testing images were carefully observed to check whether all classes were detected or not.
- Then, aspect ratios were modified in two further stages. First, aspect ratios with a small gap, such as a ratio of 1:4 between the default anchor boxes, were added. If the model produces unsuccessful results, reciprocal aspect ratios were considered. The detected images with their confidence scores were compared with the results obtained in the previous step of the analysis.
- In this way, the empirical adjustment of the scale size and aspect ratio contributed to getting the successful detection of weeds. It was further observed that the proposed method not only obtained better true positive results, but also reduced the region proposal network (RPN), classification, and localization errors.

3. Results and Discussion

3.1. Highlights of the First Stage of the Research

3.1.1. Analyzed the Performance of Several DL-Based Object Detectors

First, a comprehensive analysis of various single- and two-stage neural networks was performed in terms of percentage training loss and mean average precision. The Faster RCNN architecture achieved the highest mean average precision. It was trained with various DL backbone/classification models such as Inception-v2, ResNet-50, and ResNet-101. ResNet-101 was found to be the most suitable model that successfully extracted distinct features of seven weed classes and a negative class. The mean average precision was found to be 87.64%, and most of the weed classes were successfully detected, except for the negative class that achieved the lowest average precision (AP) of 62.35% [30].

3.1.2. Studied the Effects of Image Resizing and Interpolation Techniques

The second step was studying the effects of image resizing techniques on the deep learning models. Two image resizers were evaluated, including aspect ratio and fixed-shape resizers. Moreover, image interpolation methods were also used with both resizing techniques, including bilinear, bicubic, area, and nearest neighbor. The aspect ratio resizer with the area interpolation method was found to be the most suitable technique. This step contributed to improving the AP of the negative class to 96.61% and the mAP was also improved to 91.55%. On the other hand, this step of the work degraded the performance of the Chinee apple class to 75.78%.

3.1.3. Performance Optimization by Weight Initializers, Batch Normalization, and DL Optimizers

Finally, the weights of the Faster RCNN model were optimized. This step was divided into three stages. First, various parameters of weight initialization were investigated. Truncated normal, scaling variance, and random normal initialization techniques were used. The truncated normal and scaling variance initializers performed well, and the random normal initializer did not produce satisfactory results. The most appropriate

parameters of the truncated normal were found to be 0.01 standard deviation and zero mean; the parametric values of scaling variance were 1.0 scaling factor with the normal distribution and considered an average of the input and output weight tensor as the mode of operation.

Then, two well-known DL optimizers including RMSProp [37] and Adam [38] were analyzed in the presence and absence of batch normalization to further enhance the performance of the Faster RCNN model. It was found that the RMSProp optimizer (without batch normalization) attained the highest mAP of 93.44%. It was also observed that all classes achieved more than 90% AP, except the Chinee apple, which was degraded with an AP of 68.62%. Therefore, it was concluded that future research should attempt a higher AP of Chinee apple along with sustaining the AP of all other weed classes. Additionally, the outcomes of the former stage were validated by a stratified k-fold cross-validation technique.

Each step of the first phase of the research contributed to an improvement in mean average precision. A summary of the results obtained through the former stage of the research is presented in Figure 3.

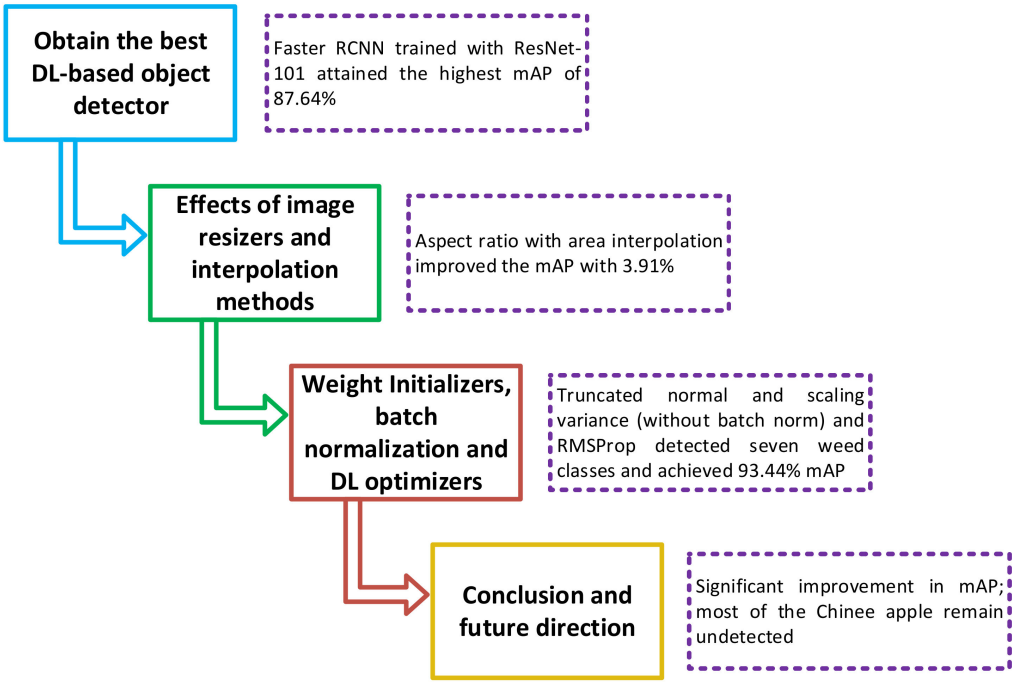


Figure 3. Highlights of the first step of the research.

3.2. Performance of the Second Phase of the Research: Enhanced Anchor Box Approach

This section presents the effects of the enhanced anchor box approach by using various combinations of scale size and aspect ratios, on the detection of weed classes and a negative class. The training performance has been analyzed by the graphical plots. The testing performance has been shown by the average precision (AP) of each class along with the mean average precision (mAP) (Table 1) along with the detection outcomes of the weed classes, to show the effectiveness of the proposed methodology.

Agronomy **2022**, *12*, 1580

Table 1. Effects of anchor scale sizes and aspect ratios on the average precision of each class.

Anchor Box Scale Sizes	Anchor Box Aspect Ratios	Average Precision (%)									mAP (%)	Training Time (h)
		Chinee Apple	Lantana	Prickly Acacia	Parthenium	Parkinsonia	Rubber vine	Siam Weed	Snake Weed	Negative		
{128^2, 256^2, 512^2}	{1:2, 1:1, 2:1}	66.39	99.83	81.23	100	99.95	98.13	87.84	24.56	49.3	78.58	10
{128^2, 256^2, 512^2}	1:1	56.11	60.12	92.81	38.11	98.33	99.84	98.94	23.62	94.9	73.63	10
{64^2, 128^2, 256^2, 512^2}	{1:2, 1:1, 2:1}	68.62	99.78	90.09	100	99.89	99.76	98.71	93.22	90.9	**93.44**	10
{32^2, 64^2, 128^2, 256^2, 512^2}	{1:2, 1:1, 2:1}	71.60	85.54	72.12	100	93.75	75.62	79.41	78.39	86.18	82.51	9.72
{16^2, 32^2, 64^2, 128^2, 256^2, 512^2}	{1:2, 1:1, 2:1}	70.39	87.38	76.11	100	94.88	72.55	86.83	79.38	85.50	83.67	9.61
{64^2, 128^2, 256^2, 512^2}	{1:4, 1:2, 1:1, 2:1}	81.65	100	65.97	100	100	100	98.95	83.92	58.95	87.71	10
{64^2, 128^2, 256^2, 512^2}	{1:4, 1:2, 1:1, 1.5:1, 2:1}	71.38	100	53.49	100	99.37	99.76	40.79	19.75	92.34	75.21	10.72
{64^2, 128^2, 256^2, 512^2}	{1:2, 1:1, 1.5:1, 2:1}	98.16	96.02	88.15	99.76	99.78	96.19	98.94	88.23	86.05	**94.58**	10.17
{64^2, 128^2, 256^2, 512^2}	{1:2, 1:1.25, 1:1, 1.5:1, 2:1}	80.94	100	48.03	100	98.91	95.67	11.82	26.35	55.51	68.58	10.22
{64^2, 128^2, 256^2, 512^2}	{1:2, 1:1, 1.25:1, 1.5:1, 2:1}	99.64	92.63	34.44	100	99.72	99.61	99.92	30.48	79.08	81.72	10.11
{64^2, 128^2, 256^2, 512^2}	{1:2, 1:1, 1.5:1, 1.75:1, 2:1}	98.58	99.11	94.13	68.79	97.74	100	94.97	51.55	94.21	88.78	10.61
{64^2, 128^2, 256^2, 512^2}	{1:2, 1:1, 1.75:1, 2:1}	100	97.25	58.4	97.7	100	97.72	37.36	72.92	64.21	80.61	10
{64^2, 128^2, 256^2, 512^2}	{1:3, 1:1, 3:1}	93.57	96.14	89.93	99.5	99.84	99.6	98.98	89.59	97.06	**96.02**	10.17
{64^2, 128^2, 256^2, 512^2}	{1:4, 1:1, 4:1}	98.64	99.72	95.16	99.56	97.79	99.08	80.86	28	89.11	87.56	10.67
{64^2, 128^2, 256^2, 512^2}	{1:3, 1:2, 1:1, 1.5:1, 2:1, 3:1}	92.63	93.8	71.02	98.1	100	99.84	99.56	92.56	86.83	92.70	10.5

16^2: 16×16, 32^2: 32×32, 64^2: 64×64, 128^2: 128×128, 256^2: 256×256, 512^2: 512×512.

Agronomy **2022**, *12*, 1580

3.2.1. Default Anchor Box Scale

During the first phase of the research, the Faster RCNN model was trained with the default settings. According to this, the model was trained with anchor box scales 64 × 64, 128 × 128, 256 × 256, and 512 × 512 with aspect ratios 1:2, 1:1, and 2:1. For this step of the research, the effects of various scales and aspect ratios have been analyzed step by step.

The authors of the original Faster RCNN architecture used several combinations of aspect ratio and scale size explained in [36]. The scale size of 128 × 128, 256 × 256, and 512 × 512 with the aspect ratios of 1:2, 1:1, 2:1, and with only 1:1, attained the highest mean average precision for that research. Similarly, both default combinations of scales and aspect ratios were applied for this study. The aspect ratio of 1:2, 1:1, and 2:1 achieved a better mean average precision, as shown in Table 1. However, the classes including Chinee apple, Snakeweed, and negative classes did not reach high average precision and were confused with other classes such as Lantana. A pictorial representation of the anchor box according to the default settings is presented in Figure 4. It can be observed that the area of the sliding window is determined by the scale size and aspect ratio. Consequently, in the case of a modification in scale and ratio, the area of the anchor box will be changed. Therefore, an important reason for an unsatisfactory result by the default settings is an unacceptable anchor box dimension that fits the characteristics/bounding box coordinates of the testing images. Hence, some of the images belonging to the testing dataset could not be detected. This reduced the mAP of the Faster RCNN model. An example of some of the classes detected by the default settings is presented in Figure 5.

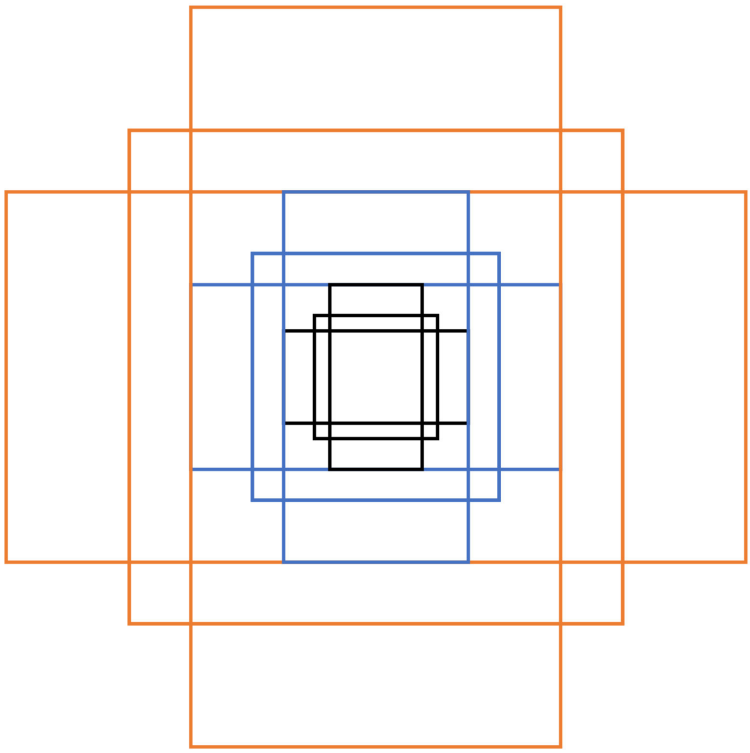


Figure 4. Anchor box with the default settings.

3.2.2. Enhancement in Anchor Box Scales

Next, smaller scale sizes were added to obtain an enhancement in the anchor boxes. In this regard, a 64 × 64-sized window was added, and its effects were evaluated. It was found that the inclusion of the 64 × 64 scale significantly improved the mAP with a huge margin of 14.86%. From the training plots presented in Figure 6a–d, both region proposal network (RPN) losses, including localization loss (R_loc_loss) and objectness loss (R_obj_loss), improved with a small difference of 0.014% and 0.104%, respectively.

Agronomy **2022**, *12*, 1580

Furthermore, losses including box classifier localization loss (Loc_class) and classification loss (Class_loss) were also reduced by 0.44% and 0.38%, respectively, compared to those obtained during training with default scale sizes, as shown in Figure 6e–h. It can be observed from Figure 6 that the Faster RCNN model did not suffer from overfitting. The loss plots were converged and settled down to the final value. Additionally, there was no such abrupt increase in the loss after achieving a steady-state condition. The anchor boxes after the addition of the 64 × 64 scale are presented in Figure 7. It can be seen that this modification in the anchor box should be useful for detecting and localizing weeds having small coordinates.

Then, the performance of the testing dataset was evaluated. The detection results for the classes including Prickly acacia, Siam weed, Snakeweed, and negatives attained a higher AP compared to the results obtained by the previous/default anchor boxes as shown in Table 1. It implies that the images belonging to these classes had coordinate ensembles on the scale of 64 × 64, resulting in a significantly higher AP. However, only Chinee apple weed was found to have low AP. There were two reasons for the insufficient results of the Chinee apple; one was confusion with other classes, including Lantana and Negative, and the second was that some images could not be detected. Examples of the four truly detected classes along with a false-positive result for the Chinee apple are presented in Figure 8.

Later, smaller scale sizes were also used to observe their effects on the Faster RCNN. For example, 32 × 32 and 16 × 16 scales were added; these anchor box scales achieved a lower mAP due to a low AP of the classes including Lantana, Prickly acacia, rubber vine, Siam weed, and Snakeweed (as shown in Table 1). However, the training time was slightly reduced with small scale sizes as compared to the default anchor box scales, as shown in Table 1. In summary, due to a substantial improvement in the performance of the Faster RCNN model after the insertion of the 64 × 64 scale, the scale sizes of 64 × 64, 128 × 128, 256 × 256, and 512 × 512 were fixed for the rest of the analysis.

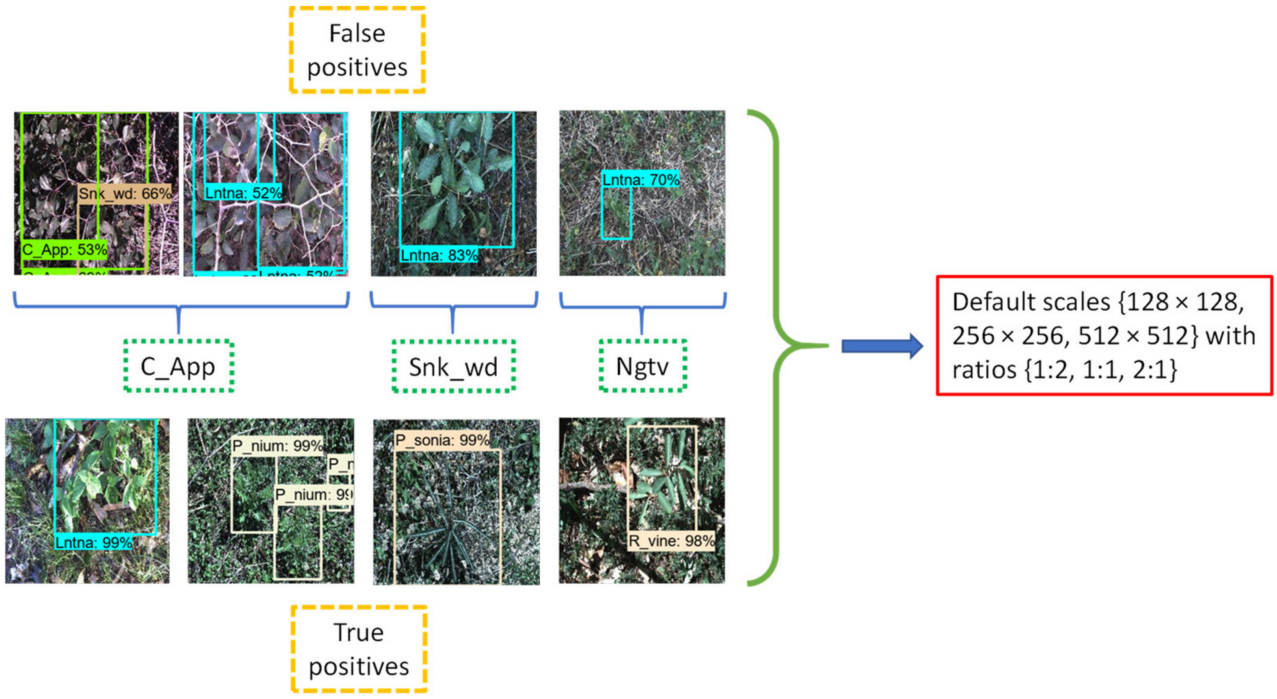


Figure 5. True-positive and false-positive results using the default scale size and aspect ratio.

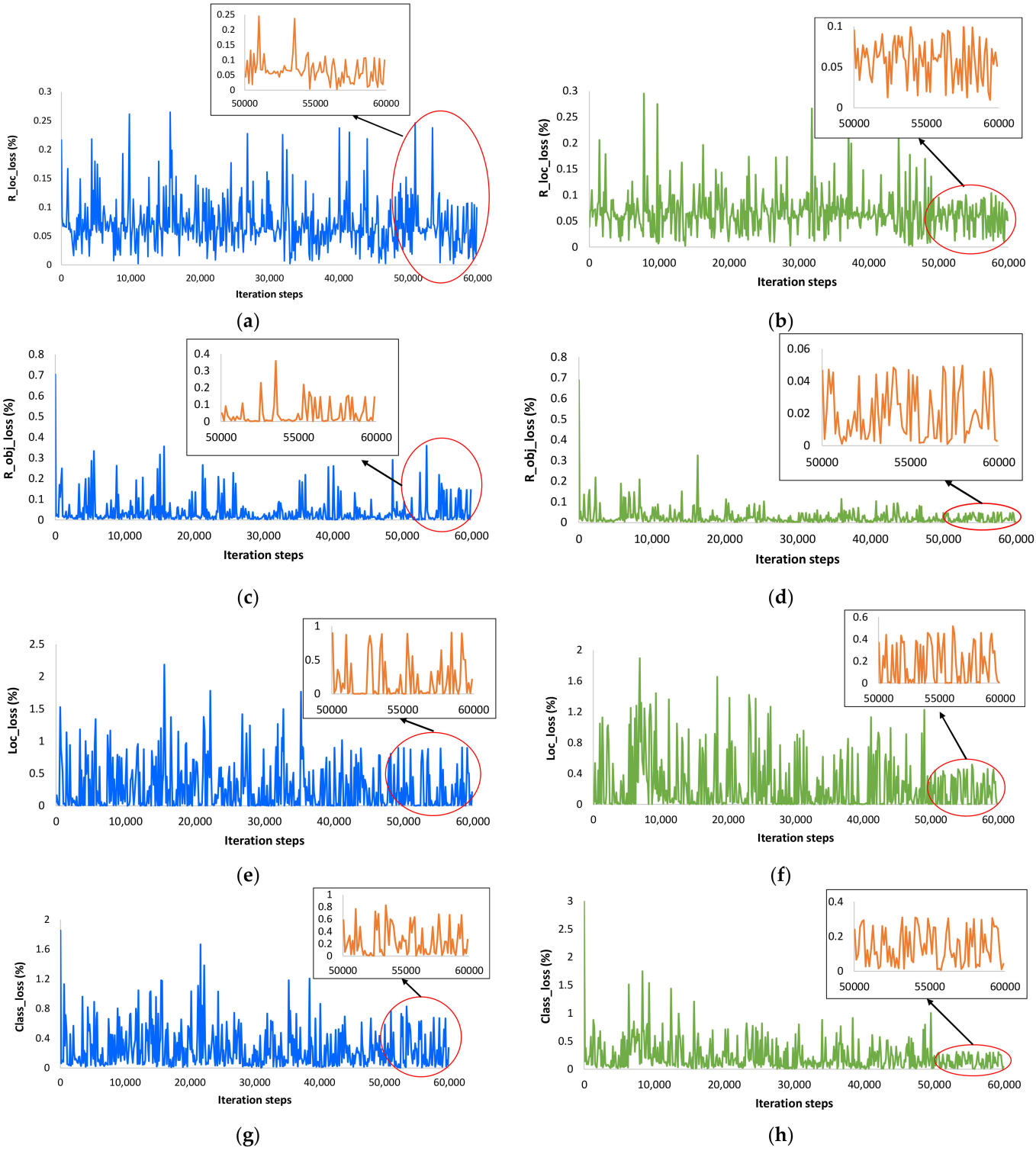


Figure 6. Training loss plots of the Faster RCNN model: (**a**) RPN localization loss with default settings; (**b**) RPN localization loss after adding 64 × 64-scale-size window; (**c**) RPN objectness loss with default settings; (**d**) RPN objectness loss after adding 64 × 64-scale-size window; (**e**) box classifier localization loss with default settings; (**f**) box classifier localization loss after adding 64 × 64-scale-size window; (**g**) box classifier classification loss with default settings; (**h**) box classifier classification loss after adding 64 × 64-scale-size window.

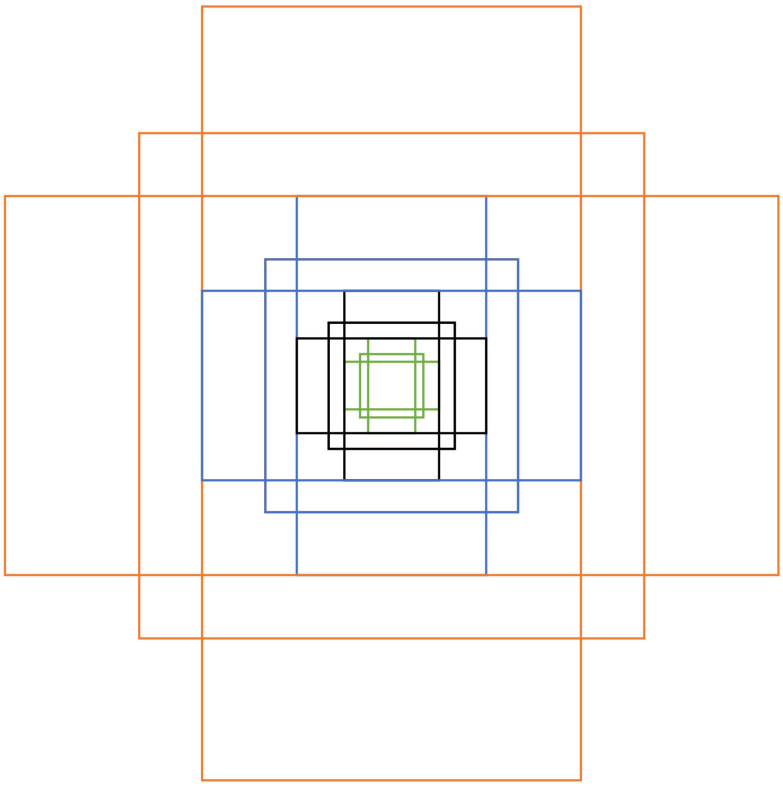


Figure 7. Enhanced anchor box after the addition of a 64 × 64 scale (green box).

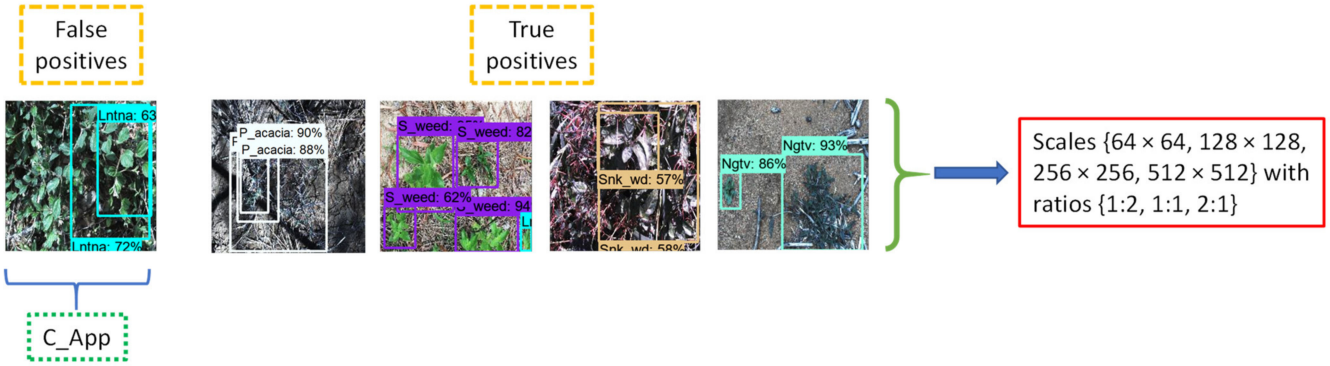


Figure 8. True positive results of Prickly acacia, Siam weed, Snakeweed, and negative class; False positive result of Chinee apple after the addition of a 64 × 64 scale.

3.2.3. Effects of Different Aspect Ratios

a. Gradual Enhancement in Aspect Ratios

The next step was to study the effects of the aspect ratio to obtain the optimum anchor boxes. First, a smaller aspect ratio of 1:4 was added to the default values. These aspect ratios marginally improved the average precision of the Chinee apple, but few classes, including Prickly acacia, Snakeweed, and negative, were degraded, as shown in Figure 9a. This was due to the mismatch of the required anchor boxes with the modified sizes. The Faster RCNN perceived (with the configurations presented in this step) that the testing images of Snakeweed and negative belonged to the Lantana. For a similar reason, Prickly acacia was confused with Parthenium.

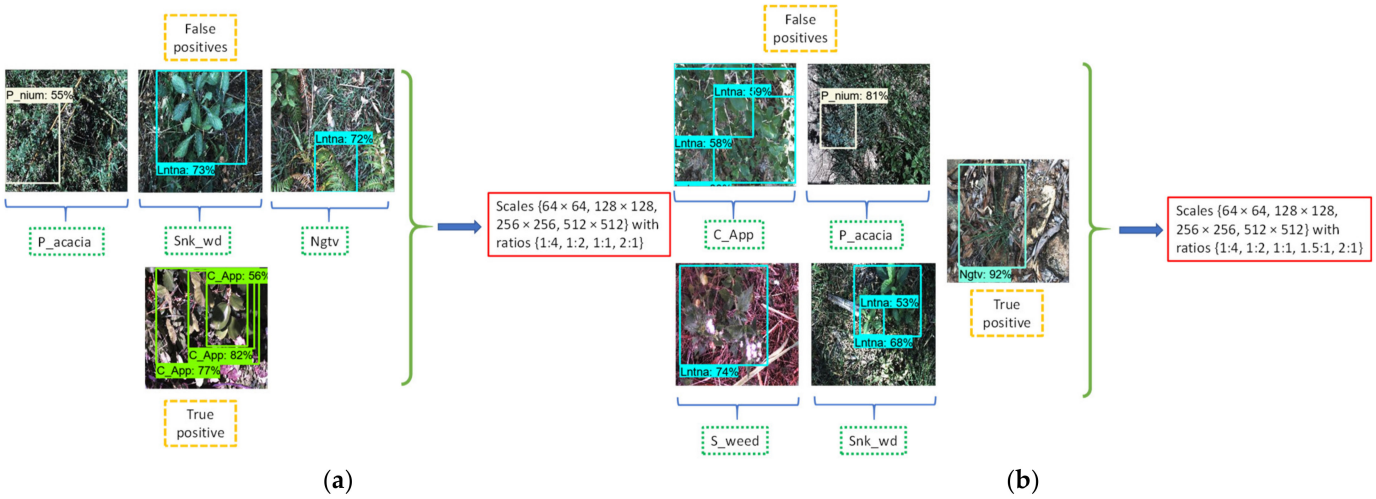


Figure 9. True and false-positive results after addition to default aspect ratios: (**a**) Results after 1:4 aspect ratio; (**b**) Results after 1:4 and 1.5:1 aspect ratios.

Then, a larger (than the original 1:1) aspect ratio was added to observe its effects on the model's performance. Therefore, an aspect ratio of 1.5:1 was added to 1:4 and default ratios, but the resulting anchor boxes were found to be not suitable for the classes of Chinee apple, Siam weed, Prickly acacia, and Snakeweed. The reason was that the addition of 1.5:1 and 1:4 aspect ratios made an anchor box that observed these classes as Lantana and Parthenium, as presented in Figure 9b. Therefore, it was proved that the simultaneous addition of a smaller and larger aspect ratio was inappropriate for the detection of weed classes.

Later, only the inclusion of an aspect ratio of 1.5:1 was examined with the default (1:2, 1:1, and 2:1). It was noticed that an addition of a 1.5:1 aspect ratio was feasible to obtain better weed detection outcomes. The Chinee apple was detected with a higher average precision of 98.16% along with a higher mAP. The successful detection results of the Chinee apple show that its images required all aspect ratios with a difference of 1:2. Moreover, four weed classes maintained their AP including Parthenium, Parkinsonia, Prickly acacia, and Siam weed. The other four classes, such as Lantana, Rubber vine, Snakeweed, and negative, achieved a lower AP. These results were acceptable since mAP was also slightly improved by 1.14% along with a significant improvement in AP of the Chinee apple. The resulting anchor boxes are presented in Figure 10.

Furthermore, it was also observed that only the box classifier localization loss (Loc_loss) improved with a margin of 0.08% compared to the loss that occurred during the previous stage of the method; the rest of the losses did not show an improvement. This small reduction in Loc_loss produced a significant improvement in the AP of the Chinee apple class. A few examples of the successful outcomes of all weed classes are presented in Figure 11.

The anchor box specifications after the addition of the 1.5:1 ratio improved the weed detection results. It shows that the required anchor box sizes can be obtained by gradual tuning of the aspect ratio. Furthermore, the correct finding of the anchor box is subjected to the empirical observations on a gradual change in anchor box scales and aspect ratios. Hence, certain intervals between anchor box enhancements were considered to identify the weeds correctly.

Few other combinations of the aspect ratio were also studied. For example, 1:1.25, 1.25:1 and 1.75:1 were added with an aspect ratio of 1.5:1 and default (1:2, 1:1, and 2:1). The aspect ratios of 1:1.25 and 1:25:1 gave a lower mAP (as shown in Table 1). For reference to their false-positive results, Figure 12a,b present the detection results. However, an aspect ratio of 1.75:1 (combined with 1.5:1 and default aspect ratios) was reasonable to detect the classes of weeds, except for the Parthenium and Snakeweed, which were confused with Prickly acacia and Chinee apple, respectively, as shown in Figure 12c. Finally, the effects of

a 1.75:1 aspect ratio with default ratios were studied. The classes including Prickly acacia, Siam weed, Snakeweed, and negative classes were seen to not detect well (Figure 12d). The training time with the addition of aspect ratios was marginally increased, as presented in Table 1.

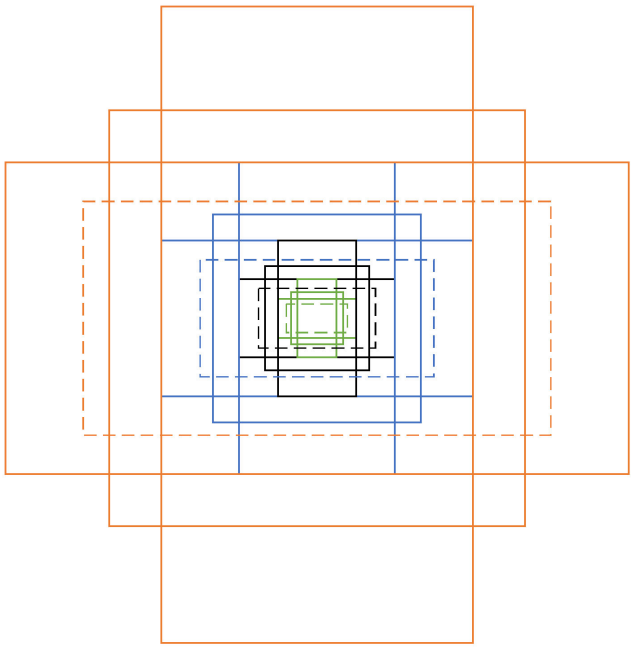


Figure 10. Resultant anchor box after adding 1.5 aspect ratio (dotted line boxes).

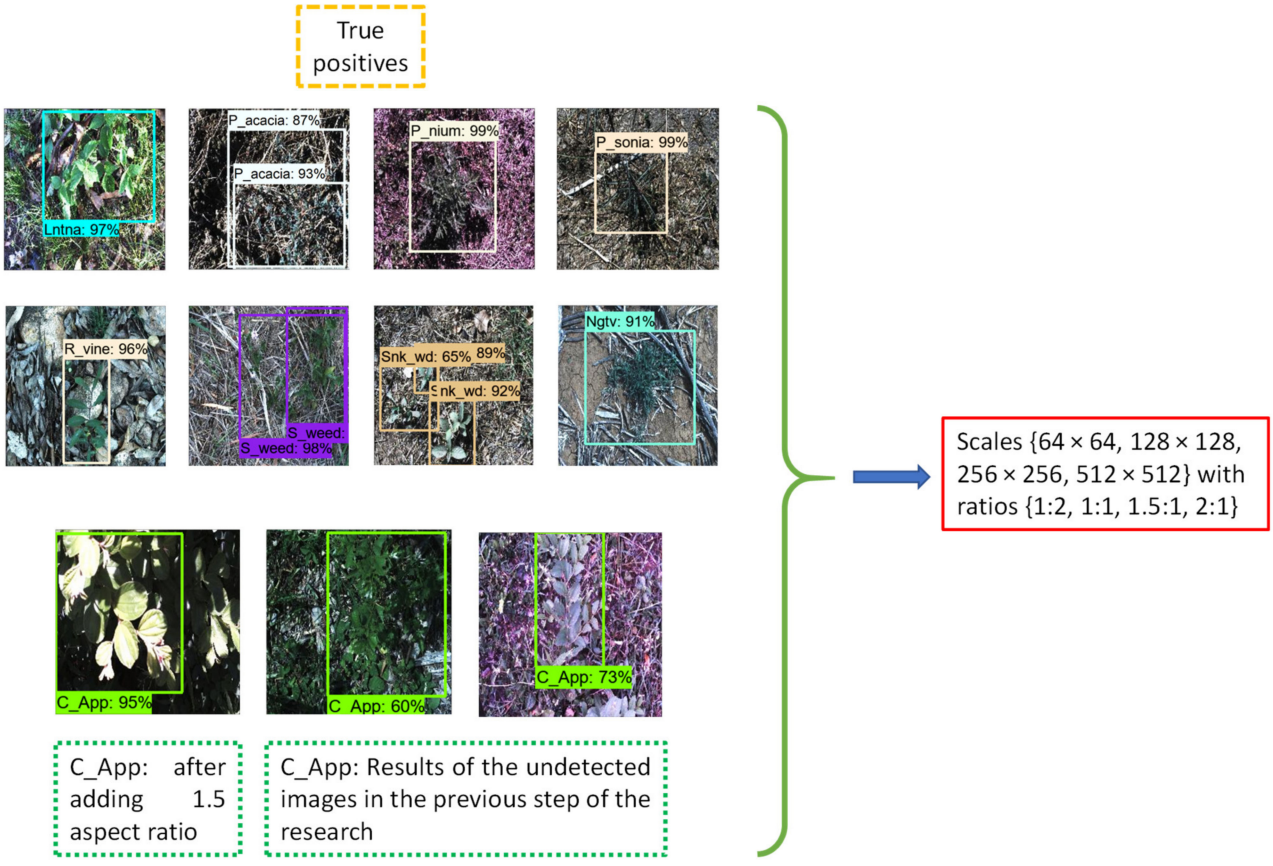


Figure 11. Detection results for all classes after the addition of a 1.5 aspect ratio.

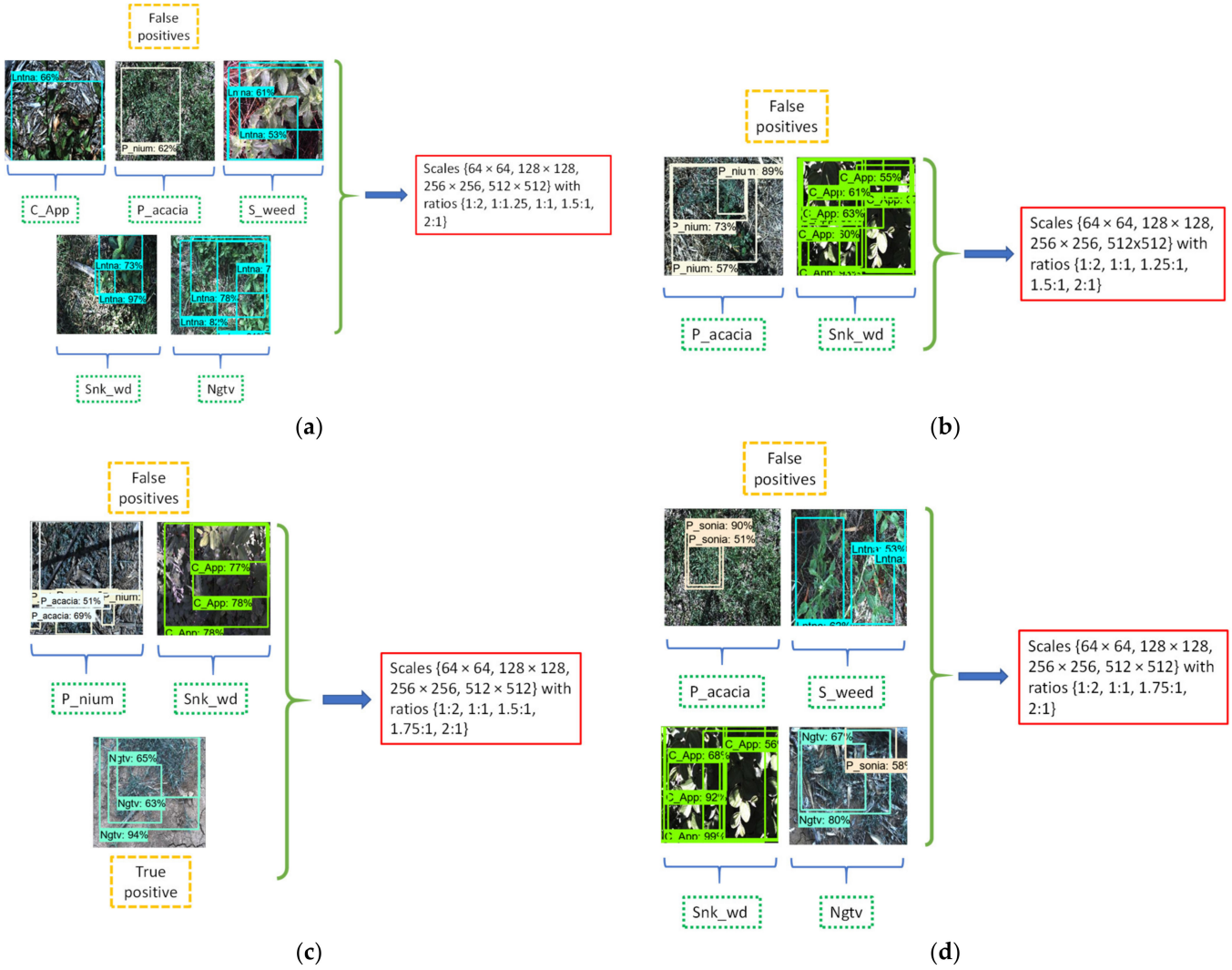


Figure 12. Detection results after the addition of various aspect ratios to the default ratios: (**a**) presents false positives with aspect ratios of 1:1.25 and 1.5:1; (**b**) presents false positives with aspect ratios of 1.25: 1 and 1.5:1; (**c**) presents true and false positives with aspect ratios of 1.5:1 and 1.75:1; (**d**) presents false positives with an aspect ratio of 1.75:1.

b. Reciprocal Enhancement in Aspect Ratios

After the addition of a 1.5:1 aspect ratio, the AP of the Chinee apple improved significantly. However, the "Negative" class was marginally degraded. From a practical perspective, the non-weed/negative class should also be detected and localized accurately. It is very useful for a site-specific weed management system when there is always a need for discrimination between a weed and a non-weed class to apply the herbicide spray precisely. Moreover, one of the main objectives of this research was to maintain the high AP of all classes. Therefore, an attempt was made to improve the AP of the negative class. In this regard, the anchor boxes were enhanced by reciprocal aspect ratios.

First, the effects of the aspect ratio of 1:3 and 3:1 (Figure 13) on the enhanced anchor scales were studied. Most training losses were reduced, including box classifier classification and localization losses, and RPN objectness loss with a margin of approximately 0.12%, 0.11%, and 0.02%, respectively, compared to the loss obtained with a gradual addition in an aspect ratio of 1.5:1, as shown in Figure 14. The training plots show that the model did not suffer from commonly occurring problems such as overfitting, as the loss value reached its steady point and there was no sudden change after obtaining the convergence. These training results were also validated by the testing outcomes; the negative class

Agronomy **2022**, *12*, 1580

achieved a significantly higher AP of 97.06% along with maintaining the AP of the classes that included Parthenium, Parkinsonia, Rubber vine, and Siam weed. The Chinee apple again attained a high AP of 93.57%. Few classes including Lantana, Prickly acacia, and Snakeweed were degraded with a bearable margin of 1–4% average precision as shown in Table 1. However, the mAP also improved to 96.02%, which was 2.58% better than the results achieved in the previous stage of the research. A few examples of all detected classes are represented in Figure 15. Furthermore, few other reciprocal combinations of aspect ratios were tested, which did not provide any significant improvements. Only ratios of 1:4 and 4:1 attained considerable outcomes, but Snakeweed was detected with only 28% AP as shown in Table 1. Finally, an anchor box with a combination of the two best results was also generated, considering the aspect ratios 1:3, 1.5:1, and 3:1, with the default ratios of 1:2, 1:1, and 2:1. The results of all weed classes with these aspect ratios were satisfactory except for prickly acacia. Therefore, the aspect ratios of 1:3 and 3:1 were found the most optimum solution to detect all the weed classes and a negative class.

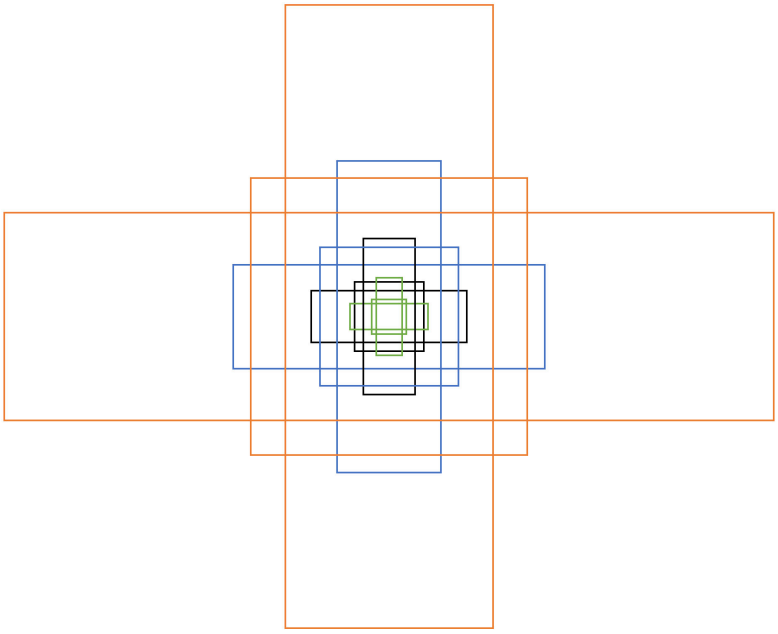


Figure 13. Enhanced anchor box after aspect ratios of 1:3 and 3:1.

The proposed anchor box method for the Faster RCNN model significantly improved the detection of weed classes, specially the Chinee apple, which was undetected/unsuccessful in the previous stage of the research. In terms of computation time, the Faster RCNN model with the default anchors required a slightly lower detection time of around 0.829 s/image compared to the final settings (with 64 × 64 scale size and 1:3, 3:1 aspect ratio) taking 0.840 s/image. However, it should be noticed that the modified Faster RCNN needed 12 anchor boxes compared to 9 anchor boxes with the default scale size and aspect ratio. Therefore, a small difference in the detection time after increasing some anchor boxes does not show any negative aspect of the proposed approach. The overall significance of the proposed method has been shown in Table 2, which compares the mean average precision with the latest DL models. It can be observed that the Faster RCNN ResNet-101 model attained the highest mAP as compared to all other DL architectures. Furthermore, it can also be seen that the performance of the model was improved by 5.8% mAP through the optimized version of the model as presented in the previous phase of the research [30]. However, the performance of the model has been further enhanced by the proposed approach in this article (enhanced anchor box approach) with a margin of 2.58% mAP. So, the overall improvement in the model's outcome was 8.38% in mAP as compared to the default settings, which shows the contribution of this research.

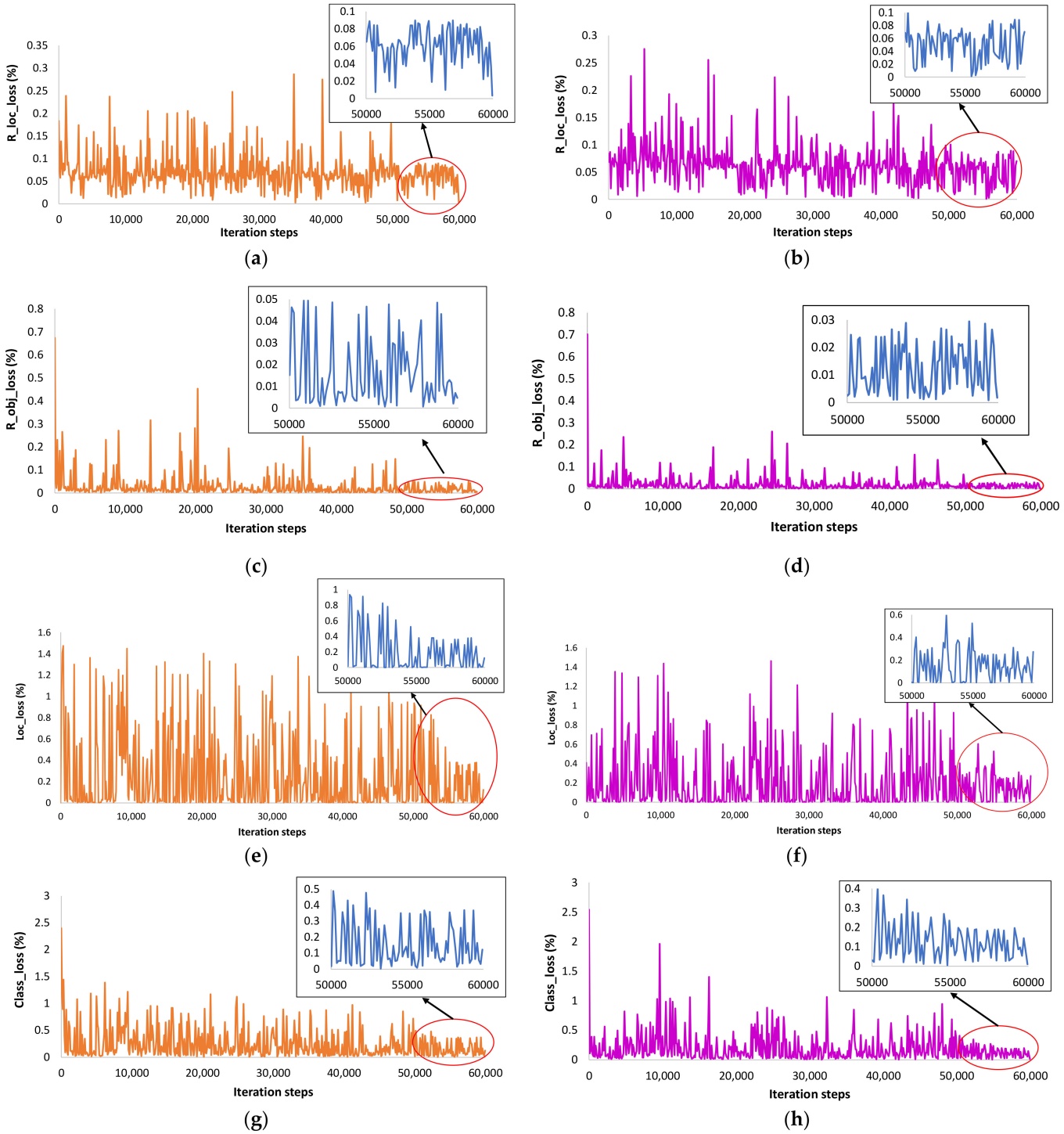


Figure 14. Training plots after adding different aspect ratios to the default: (**a**) presents the RPN localization loss with 1.5:1; (**b**) presents the RPN localization loss with 1:3 and 3:1; (**c**) presents the RPN objectness loss with 1.5:1; (**d**) presents the RPN objectness loss with 1:3 and 3:1; (**e**) presents the box classifier localization loss with 1.5:1; (**f**) presents the box classifier localization loss with 1:3 and 3:1; (**g**) presents the box classifier classification with 1.5:1; (**h**) presents the box classifier classification loss with 1:3 and 3:1.

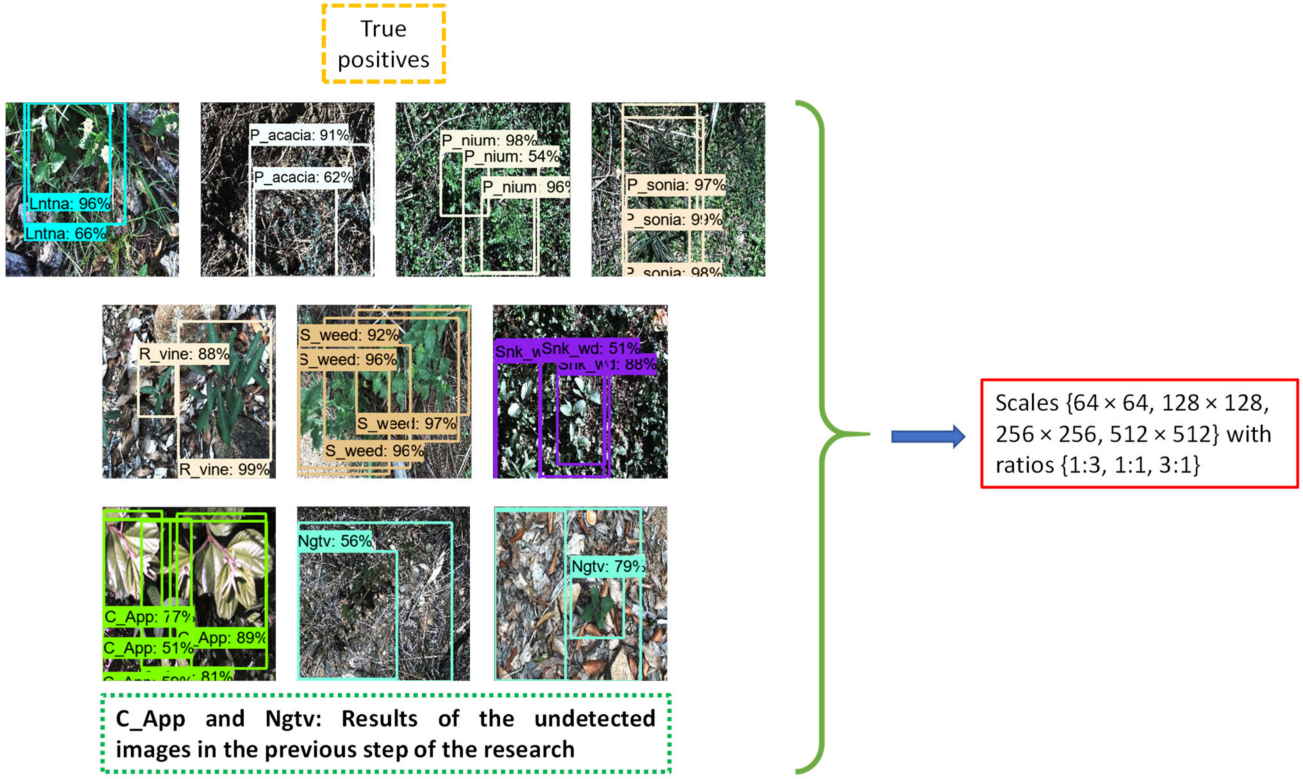


Figure 15. True-positive results with aspect ratios 1:3 and 3:1.

Table 2. Performance of the proposed method compared to different DL architectures.

DL Models with Respective Feature Extractors	Mean Average Precision (%)
YOLO-v4 CSPDarknet-53	79.68
SSD Inception-v2	48.12
SSD MobileNet	34.99
SSD ResNet-50 (RetinaNet)	21.69
EfficientDet EfficientNet	36.59
CenterNet ResNet-50	27.36
RFCN ResNet-101	55.06
Faster RCNN Inception-v2	73.59
Faster RCNN ResNet-50	87.23
Faster RCNN ResNet-101	87.64
Faster RCNN ResNet-101 (optimized model)	93.44
Faster RCNN ResNet-101 (optimized model + enhanced anchor box approach)	**96.02**

3.2.4. Validation of the Approach

Similar to the first stage of research [30], the effectiveness of this phase of the work has also been validated using a stratified five-fold cross-validation method. The final mAP obtained through the enhanced anchor box was validated in four other folds of the dataset (presented in our repository https://github.com/kmarif/DL-Weed-Identification, accessed on 20 May 2022). The presented analysis was performed in the first fold (fold1). A slight

difference in the mAP was obtained with the margin of 0.07 to 0.38% and achieved 95.95%, 96.19, 96.40%, and 95.80% by fold2, fold3, fold4, and fold5, respectively. Furthermore, the difference in AP of the Chinee apple was also not significant in all folds, with 94.27, 92.66, 92.34, and 94.95% of AP with the final enhanced anchor boxes.

To further show the robustness of the proposed research, a small external testing dataset has been generated by random google search and tested using the weights obtained by the final proposed model. The mAP is 95.83% with each class attaining AP > 90%. Figure 16 shows a sample of each detected weed and a non-weed class.

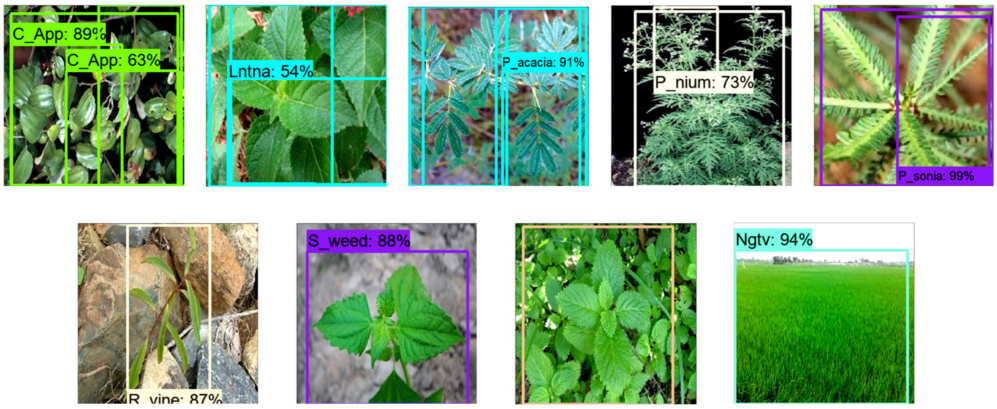


Figure 16. A sample of each class by externally generated dataset.

4. Conclusions

The architectural details of the Faster RCNN model provided a solid motivation to propose an enhanced anchor box approach to successfully identify and localize weeds. The effects of various anchor box scale sizes and aspect ratios on the training and testing performance of the model have been presented. The addition of a 64 × 64 scale size and replacement of the default aspect ratio with 1:3 and 3:1 attained the optimum results. A significant improvement of 24.95% in the average precision of the Chinee apple weed was achieved that was not successfully detected in the previous stage of the study. Furthermore, the mean average precision was 2.58% better compared to the first step of the research. In addition to that, the AP of the remaining weed classes was also maintained, which proved the effectiveness of the method. Moreover, the AP of the negative class was also improved by 6.16%. Furthermore, the successful detection results were obtained without compromising the detection time, which shows the practicality of the work. Moreover, the robustness of the work has been presented by the stratified k-fold cross-validation method and external testing dataset with a small difference of 0.38% and 0.19% in mAP, respectively. This study has provided another way to visualize and analyze deep learning-based object detectors for further development in agricultural tasks.

Although the proposed methodology has been validated by two techniques, a similar approach should also be tested on other weed classes/datasets. In the future, segmentation-based DL models can be explored to perform pixel-wise detection of weeds. Moreover, the proposed methodology for anchor box enhancement has the potential to be applied to other relevant datasets. The final weights of the proposed model could be reused for real-time weed detection in field trials using a portable system. It would be beneficial for the precise application of herbicide sprays on weeds.

Author Contributions: Conceptualization, methodology, investigation, visualization, M.H.S. and K.M.A.; software, validation, writing—original draft preparation, M.H.S.; writing—review and editing, M.H.S. and K.M.A.; project administration, supervision, funding acquisition, J.P. and K.M.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of Business, Innovation, and Employment (MBIE), New Zealand, Science for Technological Innovation (SfTI) National Science Challenge.

Data Availability Statement: The data presented in this study are made publicly available at https://github.com/kmarif/DL-Weed-Identification, (accessed on 20 May 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hasan, A.M.; Sohel, F.; Diepeveen, D.; Laga, H.; Jones, M.G. A survey of deep learning techniques for weed detection from images. *Comput. Electron. Agric.* **2021**, *184*, 106067. [CrossRef]
2. Liu, J.; Wang, X. Plant diseases and pests detection based on deep learning: A review. *Plant Methods* **2021**, *17*, 1–18. [CrossRef] [PubMed]
3. Saleem, M.H.; Potgieter, J.; Arif, K.M. Automation in agriculture by machine and deep learning techniques: A review of recent developments. *Precis. Agric.* **2021**, *22*, 2053–2091. [CrossRef]
4. Wan, S.; Goudos, S. Faster R-CNN for multi-class fruit detection using a robotic vision system. *Comput. Netw.* **2020**, *168*, 107036. [CrossRef]
5. Quiroz, I.A.; Alférez, G.H. Image recognition of Legacy blueberries in a Chilean smart farm through deep learning. *Comput. Electron. Agric.* **2020**, *168*, 105044. [CrossRef]
6. Kamilaris, A.; Prenafeta-Boldú, F.X. Deep learning in agriculture: A survey. *Comput. Electron. Agric.* **2018**, *147*, 70–90. [CrossRef]
7. Wu, Z.; Chen, Y.; Zhao, B.; Kang, X.; Ding, Y. Review of Weed Detection Methods Based on Computer Vision. *Sensors* **2021**, *21*, 3647. [CrossRef]
8. Tao, T.; Wei, X. A hybrid CNN–SVM classifier for weed recognition in winter rape field. *Plant Methods* **2022**, *18*, 1–12. [CrossRef]
9. Jiang, H.; Zhang, C.; Qiao, Y.; Zhang, Z.; Zhang, W.; Song, C. CNN feature based graph convolutional network for weed and crop recognition in smart farming. *Comput. Electron. Agric.* **2020**, *174*, 105450. [CrossRef]
10. Hu, K.; Coleman, G.; Zeng, S.; Wang, Z.; Walsh, M. Graph weeds net: A graph-based deep learning method for weed recognition. *Comput. Electron. Agric.* **2020**, *174*, 105520. [CrossRef]
11. Teimouri, N.; Jørgensen, R.N.; Green, O. Novel Assessment of Region-Based CNNs for Detecting Monocot/Dicot Weeds in Dense Field Environments. *Agronomy* **2022**, *12*, 1167. [CrossRef]
12. Yu, J.; Sharpe, S.M.; Schumann, A.W.; Boyd, N.S. Detection of broadleaf weeds growing in turfgrass with convolutional neural networks. *Pest Manag. Sci.* **2019**, *75*, 2211–2218. [CrossRef]
13. Ahmad, A.; Saraswat, D.; Aggarwal, V.; Etienne, A.; Hancock, B. Performance of deep learning models for classifying and detecting common weeds in corn and soybean production systems. *Comput. Electron. Agric.* **2021**, *184*, 106081. [CrossRef]
14. Zhuang, J.; Li, X.; Bagavathiannan, M.; Jin, X.; Yang, J.; Meng, W.; Li, T.; Li, L.; Wang, Y.; Chen, Y. Evaluation of different deep convolutional neural networks for detection of broadleaf weed seedlings in wheat. *Pest Manag. Sci.* **2021**, *78*, 521–529. [CrossRef]
15. Gao, J.; French, A.P.; Pound, M.P.; He, Y.; Pridmore, T.P.; Pieters, J.G. Deep convolutional neural networks for image-based Convolvulus sepium detection in sugar beet fields. *Plant Methods* **2020**, *16*, 1–12. [CrossRef]
16. Jin, X.; Che, J.; Chen, Y. Weed identification using deep learning and image processing in vegetable plantation. *IEEE Access* **2021**, *9*, 10940–10950. [CrossRef]
17. Espejo-Garcia, B.; Mylonas, N.; Athanasakos, L.; Fountas, S.; Vasilakoglou, I. Towards weeds identification assistance through transfer learning. *Comput. Electron. Agric.* **2020**, *171*, 105306. [CrossRef]
18. Chen, D.; Lu, Y.; Li, Z.; Young, S. Performance evaluation of deep transfer learning on multi-class identification of common weed species in cotton production systems. *Comput. Electron. Agric.* **2022**, *198*, 107091. [CrossRef]
19. Suh, H.K.; Ijsselmuiden, J.; Hofstee, J.W.; van Henten, E.J. Transfer learning for the classification of sugar beet and volunteer potato under field conditions. *Biosyst. Eng.* **2018**, *174*, 50–65. [CrossRef]
20. Bah, M.D.; Hafiane, A.; Canals, R. Deep learning with unsupervised data labeling for weed detection in line crops in UAV images. *Remote Sens.* **2018**, *10*, 1690. [CrossRef]
21. de Camargo, T.; Schirrmann, M.; Landwehr, N.; Dammer, K.-H.; Pflanz, M. Optimized Deep Learning Model as a Basis for Fast UAV Mapping of Weed Species in Winter Wheat Crops. *Remote Sens.* **2021**, *13*, 1704. [CrossRef]
22. Ukaegbu, U.F.; Tartibu, L.K.; Okwu, M.O.; Olayode, I.O. Development of a Light-Weight Unmanned Aerial Vehicle for Precision Agriculture. *Sensors* **2021**, *21*, 4417. [CrossRef]
23. Veeranampalayam Sivakumar, A.N.; Li, J.; Scott, S.; Psota, E.; Jhala, A.J.; Luck, J.D.; Shi, Y. Comparison of object detection and patch-based classification deep learning models on mid-to late-season weed detection in UAV imagery. *Remote Sens.* **2020**, *12*, 2136. [CrossRef]
24. Khan, S.; Tufail, M.; Khan, M.T.; Khan, Z.A.; Anwar, S. Deep learning-based identification system of weeds and crops in strawberry and pea fields for a precision agriculture sprayer. *Precis. Agric.* **2021**, *22*, 1711–1727. [CrossRef]
25. Ruigrok, T.; van Henten, E.; Booij, J.; van Boheemen, K.; Kootstra, G. Application-specific evaluation of a weed-detection algorithm for plant-specific spraying. *Sensors* **2020**, *20*, 7262. [CrossRef]
26. Kounalakis, T.; Triantafyllidis, G.A.; Nalpantidis, L. Deep learning-based visual recognition of rumex for robotic precision farming. *Comput. Electron. Agric.* **2019**, *165*, 104973. [CrossRef]
27. Khan, A.; Ilyas, T.; Umraiz, M.; Mannan, Z.I.; Kim, H. Ced-net: Crops and weeds segmentation for smart farming using a small cascaded encoder-decoder architecture. *Electronics* **2020**, *9*, 1602. [CrossRef]

28. Wang, A.; Xu, Y.; Wei, X.; Cui, B. Semantic segmentation of crop and weed using an encoder-decoder network and image enhancement method under uncontrolled outdoor illumination. *IEEE Access* **2020**, *8*, 81724–81734. [CrossRef]
29. Zou, K.; Chen, X.; Wang, Y.; Zhang, C.; Zhang, F. A modified U-Net with a specific data argumentation method for semantic segmentation of weed images in the field. *Comput. Electron. Agric.* **2021**, *187*, 106242. [CrossRef]
30. Saleem, M.H.; Velayudhan, K.K.; Potgieter, J.; Arif, K.M. Weed Identification by Single-Stage and Two-Stage Neural Networks: A Study on the Impact of Image Resizers and Weights Optimization Algorithms. *Front. Plant Sci.* **2022**, *13*, 920. [CrossRef]
31. Olsen, A.; Konovalov, D.A.; Philippa, B.; Ridd, P.; Wood, J.C.; Johns, J.; Banks, W.; Girgenti, B.; Kenny, O.; Whinney, J. DeepWeeds: A multiclass weed species image dataset for deep learning. *Sci. Rep.* **2019**, *9*, 2058. [CrossRef] [PubMed]
32. Saleem, M.H.; Khanchi, S.; Potgieter, J.; Arif, K.M. Image-based plant disease identification by deep learning meta-architectures. *Plants* **2020**, *9*, 1451. [CrossRef] [PubMed]
33. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In *Computer Vision—ECCV 2014, Proceedings of the 13th European conference, Zurich, Switzerland, 6–12 September 2014*; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Springer: Cham, Switzerland, 2014; pp. 740–755.
34. Bergstra, J.; Bengio, Y. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* **2012**, *13*, 281–305.
35. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
36. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [CrossRef] [PubMed]
37. Hinton, G.; Srivastava, N.; Swersky, K. Neural networks for machine learning. *Coursera Video Lect.* **2012**, *264*, 2146–2153.
38. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980. Available online: https://arxiv.org/abs/1412.6980 (accessed on 20 May 2022).

Chapter 7 A Performance-Optimized Deep Learning based Plant Disease Detection Approach for Horticultural Crops of New Zealand

This chapter contains content from the following article.

M. H. Saleem, J. Potgieter, and K. M. Arif, "A performance-optimized deep learning-based plant disease detection approach for horticultural crops of New Zealand," *IEEE Access*, vol. 10, pp. 89798-89822, 2022. http://doi.org/10.1109/ACCESS.2022.3201104

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 28 July 2022, accepted 10 August 2022, date of publication 23 August 2022, date of current version 31 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3201104

RESEARCH ARTICLE

A Performance-Optimized Deep Learning-Based Plant Disease Detection Approach for Horticultural Crops of New Zealand

MUHAMMAD HAMMAD SALEEM[1], JOHAN POTGIETER[2], AND KHALID MAHMOOD ARIF[1], (Senior Member, IEEE)

[1]Department of Mechanical and Electrical Engineering, School of Food and Advanced Technology, Massey University, Auckland 0632, New Zealand
[2]Massey AgriFood Digital Laboratory, Massey University, Palmerston North 4472, New Zealand

Corresponding author: Khalid Mahmood Arif (k.arif@massey.ac.nz)

This work was supported by the Ministry of Business, Innovation, and Employment (MBIE), New Zealand, Science for Technological Innovation (SfTI) National Science Challenge.

ABSTRACT Deep learning-based plant disease detection has gained significant attention from the scientific community. However, various aspects of real horticultural conditions have not yet been explored. For example, the disease should be considered not only on leaves, but also on other parts of plants, including stems, canes, and fruits. Furthermore, the detection of multiple diseases in a single plant organ at a time has not been performed. Similarly, plant disease has not been identified in various crops in the complex horticultural environment with the same optimized/modified model. To address these research gaps, this research presents a dataset named NZDLPlantDisease-v1, consisting of diseases in five of the most important horticultural crops in New Zealand: kiwifruit, apple, pear, avocado, and grapevine. An optimized version of the best obtained deep learning (DL) model named region-based fully convolutional network (RFCN) has been proposed to detect plant disease using the newly generated dataset. After finding the most suitable DL model, the data augmentation techniques were successively evaluated. Subsequently, the effects of image resizers with interpolators, weight initializers, batch normalization, and DL optimizers were studied. Finally, performance was enhanced by empirical observation of position-sensitive score maps and anchor box specifications. Furthermore, the robustness/practicality of the proposed approach was demonstrated using a stratified k-fold cross-validation technique and testing on an external dataset. The final mean average precision of the RFCN model was found to be 93.80%, which was 19.33% better than the default settings. Therefore, this research could be a benchmark step for any follow-up research on automatic control of disease in several plant species.

INDEX TERMS Convolutional neural networks, deep learning, optimization algorithms, cross-validation, plant disease detection.

I. INTRODUCTION

According to the latest fresh facts report by a New Zealand (NZ) research and development organization named Plant and Food Research, the horticultural industry achieved a record export of over NZ$6.6 billion by June 2020 [1]. The most prominent fresh fruits were kiwifruit, apples, and avocadoes with an export value of NZ$2.5, NZ$0.9, and NZ$0.1 billion, respectively, followed by the New Zealand wine at NZ$1.9 billion [1]. Furthermore, the largest crop area of 39,935 ha has been estimated for grapevines, whereas 12,905 ha of kiwifruit and 10,750 ha of apples, pears, and nashi have been reported. Based on these statistics, horticultural crops generate a great impact on New Zealand's economy. Hence, addressing the problems associated with horticultural crops could further strengthen the export value of the horticultural sector.

Among several real field problems, plant diseases affect crop yield, and quality [2], and cause economic losses [3]. The precise detection of the disease is an important step

The associate editor coordinating the review of this manuscript and approving it for publication was Turgay Celik.

in reducing its spread to neighboring plants, application of appropriate disease control treatments, and improving crop productivity. In this regard, this research is dedicated to the accurate identification of plant diseases by deep learning (DL) in the most valuable crops of NZ, including kiwifruit, apple, pear, avocado, and grapevine. Furthermore, several research gaps related to the dataset, real horticultural conditions, and deep learning-based plant disease recognition have been addressed in this study.

Deep learning, a subset of machine learning, has been reported in literature as a successful technique in recognizing plant diseases. A recent review summarized and compared various pre-processing steps (image resizing, data augmentation, normalization and standardization, data annotation, and others), datasets, convolutional neural networks, training techniques, deep learning frameworks, and optimization algorithms [4]. Another review article presented various modified DL models, plant disease detection tasks, problems, and challenges of DL-based plant disease classification [5]. For instance, the significance of the recent solutions for the small datasets was presented, such as transfer learning, few-shot learning, and one-shot learning. Furthermore, the early plant detection problem was explained with the use of hyperspectral imaging (HSI).

In the early stages of research in DL-based plant disease identification, the major focus remained on the classification tasks. For example, [6] did the early work in the domain of DL-based recognition and classification of plant disease by using two well-known DL models, namely AlexNet and GoogLeNet. Similarly, the task of the classification of plant diseases was presented in various articles by transfer learning and fine-tuning methods [7], [8]. These articles showed the importance of using the latest training techniques.

In the next stage, research community focused on the dataset size, particularly small datasets, as it played a significant role in the performance of the DL models. A novel data augmentation technique to classify disease in cassava leaves, tested on a modified MobileNet model, was presented in [9]. Another research [10] presented a generative adversarial network (GAN) for classifying the disease in PlantVillage dataset [11], which contains 38 classes of healthy/disease leaves for 14 plant species. Yet another article discussed a GAN-based model to classify tomato leave disease [12]. These articles formed a basis for data augmentation in plant disease detection. However, they only studied the performance of the DL models on single datasets leaving questions around their performance with other datasets containing disease in different crops.

Modification of well-known DL models is another area of research that has seen continuous focus for a long time. A modified CenterNet model with DenseNet-77 was proposed by [13] to identify plant disease while a MobileNet was modified for the classification of plant disease by [14].

A research showing the effectiveness of deep learning optimizers was presented in [15]. Then, a study was focused on the plant disease identification task that contains both classification and localization in a single framework, using the same PlantVillage dataset [16]. Although, this research presented an improvement in the accuracy of plant disease detection and classification tasks, the major limitation was the analysis of the deep learning technology in a controlled environment dataset.

Some of the studies also focused on datasets collected in a real agricultural environment. For example, an article presented tomato disease detection in real agricultural conditions, using three DL object detectors named Faster Region-based Convolutional Neural Network (RCNN), Single-shot MultiBox Detector (SSD), and Region-based Fully Convolutional Network (RFCN) [17]. Various real-world scenarios were considered. However, the external dataset could also be tested to validate the research. A study presented a Convolutional Neural Network (CNN) named SoyNet, to classify the disease in soybean leaves after segmenting the images of leaves [18]. This study presented variations in the parameters of the DL model, such as dropout, pooling operations, and inclusion of activation functions. These adjustments were found to be successful in improving the performance of the model. Moreover, the usefulness of the proposed method was compared with other techniques. An article presented the classification of cardamom plant diseases using the EfficientNet-V2 model [19]. This study did not provide training profiles/plots. A multilayer convolutional neural network was presented for the classification of disease on mango leaves [20]. Although this article presented the significance of the DL model compared with other machine learning-based techniques, better effectiveness should have been shown by comparing it with the DL models as well. In recent research, an improved version of the Xception model was proposed for the identification of peach diseases [21]. The novelty of this work was shown by comparing the proposed method with the well-known models. However, other modified versions of the state-of-the-art DL models could be used for the analysis. Tomato disease were detected using a modified version of the you look only once (YOLO) model [22]. It was observed that the training performance of the models was presented with limited information. Another study presented a DL-based method for the detection of tomato disease divided into target and control classes [23]. This research proposed a new way of performing plant disease detection task that can open various opportunities for future research. An improved region proposal network was proposed for the detection of northern maize leaf blight [24]. A few studies have also proposed real-time detection of plant disease. A DL model was presented for the identification of tomato disease [25]. Similarly, disease detection on grape leaves was performed using a DL architecture based on Faster region-based convolutional neural network (R-CNN) [26].

After rapid advancement and research on deep learning-based plant disease identification, there are still important research gaps and considerable room for further developments to investigate the practical aspects of horticultural fields. The current literature has mainly focused on the plant

disease detection task to only plant leaves. Moreover, the available datasets emphasize on the presence of a single disease at a time in a plant leaf. Furthermore, recent studies have shown high accuracy in the PlantVillage dataset (which contains defective leaves in 14 plant species) [16], but none of the articles have provided the significance of a single deep learning model for different crops in real agricultural conditions. This is important to consider, as each crop could have different background elements. Therefore, the robustness of the DL should be analyzed for that case.

This research addresses several research questions related to the capability of DL to address various complex agricultural problems. The first question is whether deep learning can perform plant disease detection with the same trained/optimized/modified model for three problems at a time: (a) identification of diseases in several organs of plants, (b) presence of multiple diseases in a plant organ, and (c) recognition of diseases in various crops considering variations in their environments/background elements? Connecting to the previous question, can a DL architecture correctly distinguish symptomatically identical diseases in different crops? The final question is how well can the attained accuracy of the DL-based method be validated for the problems highlighted earlier?

To answer these questions, this article presents a deep learning-based performance optimization approach. First, the dataset images were collected from various New Zealand farms and horticultural fields. It contains 20 classes of healthy and defective leaves, fruits, and stems/canes of five different crops. Then, various image resizing techniques, batch normalization, and weight optimization were applied. These techniques have not yet been explored for the detection of plant disease. Furthermore, the main novelty of the RFCN model (the most suitable DL architecture obtained after comparing several models) was analyzed. In this regard, the position-sensitive score maps were empirically evaluated, and anchor boxes were modified, to obtain high accuracy for the identification of all healthy and disease classes.

This research also addresses some of the research gaps outlined in recent articles, such as the validity of deep-learning-based plant disease identification. Moreover, the data augmentation has been applied after dividing the data into training, validation, and testing sub-datasets, to avoid biased results; otherwise, there was a possibility to get similar images in the sub-datasets. Furthermore, this study provided new insights into DL-based plant disease detection, rather than giving redundant discussions using the excessively explored dataset like PlantVillage [27].

The key contributions of this research are:

1) A new dataset of plant diseases has been proposed for the most important horticultural crops in New Zealand, named NZDLPlantDisease-v1.
2) Detection of disease has been performed in multiple plant organs for five different crops.
3) The presence and detection of multiple diseases on a single plant organ have been addressed.
4) The effects of data augmentation techniques have been studied by dividing them into various categories rather than considering all conventional methods together.
5) A comprehensive deep learning-based plant disease detection pipeline has been presented. In this regard, various steps have been explored prior to suggesting any modification to the state-of-the-art DL models.
6) The confusion/false positive results in symptomatically similar diseases (occurring in different crops) have been addressed. An in-depth analysis of the best-obtained DL model named region-based fully convolution network (RFCN) has been performed by position-sensitive score maps and anchor boxes.
7) The proposed approach has been validated using a stratified k-fold cross-validation technique and an external testing dataset.

II. MATERIALS AND METHODS

A. PROPOSED APPROACH

The proposed methodology consists of various practical considerations related to the presence of plant diseases in a real horticultural environment. A comprehensive deep learning-based optimization approach has been proposed. The presented methodology has successfully solved three identified agricultural problems, including the detection of disease in multiple plant organs, the identification of disease in different crops, and the presence of multiple diseases in a plant organ at a time. These problems have been solved by different techniques presented in sub-sections. The idea was to improve the average precision of each class. The results from each of the step were evaluated, the respective problems were highlighted and addressed in the next step.

First, the research questions were outlined to begin collecting images of the dataset. Next, well-known DL architectures were compared, and the two best deep learning (DL) models were obtained, which attained the highest mean average precision. Then, the data augmentation techniques were applied category-wise, including color change (brightness, contrast, and sharpness), the inclusion of noise with variation in color, rotational and translational changes, and finally, the combination of all categories, including the original images. The next step was the performance optimization of the DL model using various techniques. In this regard, the effects of image resizers and interpolators were analyzed. This step was performed to investigate different input images for the DL-based plant disease detection. Then, different DL initializers were tested. Subsequently, batch normalization was applied to cope with the internal covariate shift. Then, DL optimizers were leveraged to optimize the weights of the deep learning model. This led to a further improvement in the performance of classes that achieved low average precision (AP). This objective was achieved by empirically analyzing the novelty of the best-obtained model. The final step was the modification of the DL model by empirically tuning its anchor box scale and aspect ratios. In case of unsatisfactory results, the feature extractor/classification model had to be modified.

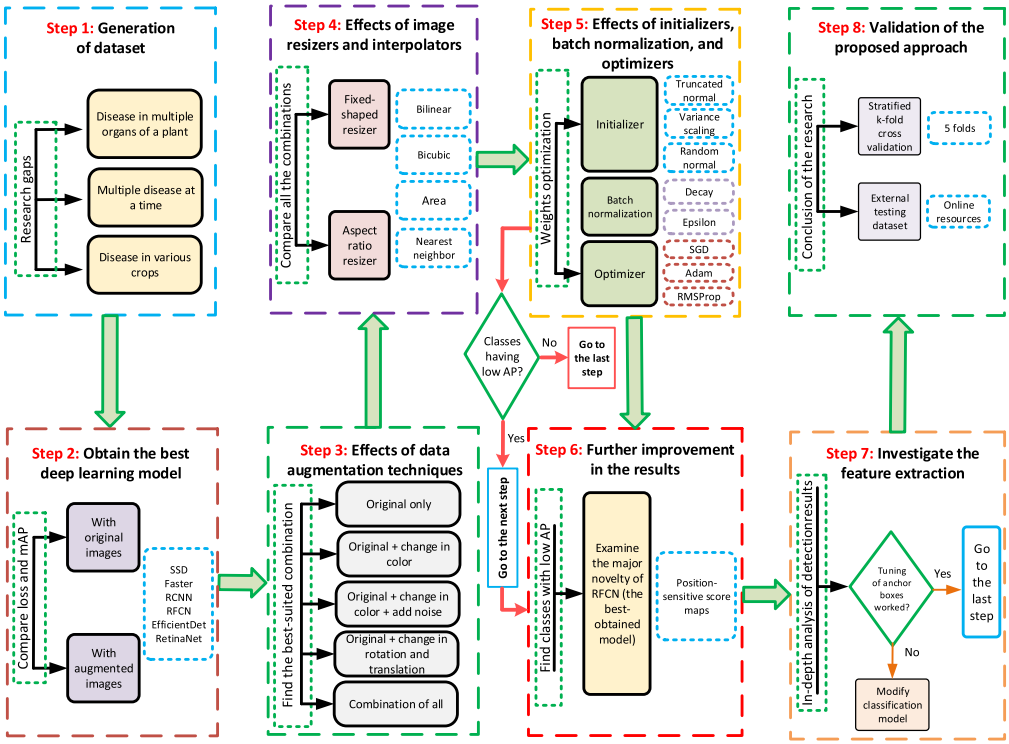


FIGURE 1. The overall workflow of the proposed methodology.

The final results were validated using a stratified k-fold cross-validation technique and a test dataset generated through various online/open-source images. The overall methodology of this study is presented in Fig. 1

B. NZDLPLANTDISEASE-V1 DATASET

1) OVERVIEW AND GENERAL INFORMATION OF THE PROPOSED DATASET

The proposed dataset has several properties of real agricultural fields that have not been presented in previous open-source datasets. A comprehensive overview of several datasets along with the new/proposed dataset for this research is presented in Table 1. Further details of the important features of the presented dataset are explained in the following subsections.

This dataset contains plant disease in five different crops in New Zealand, including kiwifruit, apple, pear, avocado, and grapevine, named NZDLPlantDisease-v1. The images were acquired by using a Samsung smartphone Galaxy S10 plus: 12 MP f/1.5-2.4 (wide), 12 MP f/2.4 (telephoto), and 16 MP, f/2.2 (ultrawide). Several local horticultural fields were visited in Auckland and Palmerston North, New Zealand. The images were taken at a working distance of 200-300 mm.

2) PRACTICAL CONSIDERATIONS

The dataset was collected between December 2020 and May 2021. The abrupt change in New Zealand's weather was considered a positive aspect of the dataset generation because it helped obtain diversity in the dataset via variations in illumination and environmental conditions. Furthermore, dataset images were captured in the presence and absence of shadows to include real horticultural conditions. Several examples of these practical considerations are shown in Fig. 2.

FIGURE 2. Examples of healthy and defected leaves of kiwifruit and apple in the presence and absence of shadow.

3) MULTI-DISEASE AND MULTI-ORGAN DATASET IMAGES

One of the research gaps addressed in this article is the detection of the disease in various organs/parts of the plants. Therefore, healthy and disease classes are considered in the leaves, stems, and fruits of apple and pear. However, the dataset classes for avocado and kiwifruit only consist of leaves. The images for the grapevine were only taken for healthy and disease cane, due to the end of the season of the grapevines at the time of dataset collection. In this research, the presence of multiple classes of disease in plant organs has also been addressed. For example, black spot, mosaic virus, and glomerella leaf spot (or two of them) were present in some of the apple leaves at one time. Similarly, algal leaf spot

TABLE 1. Overview of different plant disease datasets.

Datasets	Plant Organs	Number of healthy and disease classes	Main features
Cassava disease **[28]**	Leaves	5	Real agricultural conditions, presence, and absence of shadows
Rice disease **[29]**	Leaves	4	Real agricultural environment
Citrus Dataset **[30]**	Leaves and fruits	10	Both plain and real background
Maize disease **[31]**	Leaves	1	Dataset images in three different ways: a hand-held camera, a camera mounted on a UAV, and a boom. Annotations are also provided. The largest open-source dataset for any single plant leaf disease.
Grapevine [5]	Leaves	2	Real background
Arabica coffee **[32]**	Leaves	5	Cropped dataset, emphasized the region of interest
PlantVillage **[11]**	Leaves	38	Controlled environment, contains the highest number of plant disease/healthy classes
NZDLPlant Disease-v1	Leaves, stem/cane, and fruits	20	Real agricultural conditions, disease in three plant organs, five different crops, the occurrence of multiple diseases in a plant organ at a time, and various conditions like different illumination conditions, presence/absence of shadows, environmental changes

and branch canker were present on avocado leaves at the same time. Samples of multiple disease problems are presented in Fig. 3.

FIGURE 3. Examples of multiple disease problems: (a) shows the apple mosaic virus and the glomerella leaf spot; (b) shows the apple mosaic virus and the black spot; (c) shows the avocado algal leaf spot and the branch canker.

4) ANNOTATIONS OF HEALTHY AND DISEASE CLASSES

The number of images from each class ranged from 60 to 318, as presented in Table 2. The NZDLPlantDisease-v1 dataset

TABLE 2. Summary of NZDLPlantDisease-v1 dataset.

Crops	Plant Organs	Classes of NZDLPlantDisease-v1	Pathogens	Annotation labels
Apple	Leaf	Black spot (scab)- Venturia inaequalis (81)	Fungi	A_blk_spot
		Glomerella leaf spot - Glomerella cingulate (271)	Fungi	A_gl_lf_spot
		Mosaic virus (AMV) (224)	Virus	A_m_virus
		Healthy (215)	-	A_healthy_l
	Stem	European canker - Neonectria ditissima (169)	Fungi	A_e_canker
	Fruit	Black rot - Botryosphaeria obtuse (65)	Fungi	A_blk_rot_f
		Healthy (177)	-	A_healthy_f
Avocado	Leaf	Algal leaf spot - Cephaleuros virescens (104)	Algae	Av_alg_lf_spot
		Branch canker - Botryosphaeria spp. (250)	Fungi	Av_br_canker
		Healthy (226)	-	Av_healthy_l
Grapevine	Cane	Black spot - Elsinoe ampelina (80)	Fungi	G_blk_spot_c
		Healthy (170)	-	G_healthy_c
Kiwifruit	Leaf	Bacterial canker - Pseudomonas syringae pv. Actinidiae (318)	Bacteria	Kf_bac_canker
		Healthy (271)	-	Kf_healthy_l
Pear	Leaf	Fire blight - Erwinia amylovora (123)	Bacteria	P_fr_blight
		Scab - Venturia pirina (208)	Fungi	P_scab
		Healthy (215)	-	P_healthy_l
	Stem	European canker - Neonectria ditissima (96)	Fungi	P_canker
	Fruit	Stony pit (73)	Virus	P_s_pit
		Healthy (209)	-	P_healthy_f

Parentheses indicate the number of images, and the hyphens (-) in the pathogen column indicate healthy classes.

was divided into three sub-datasets: training (70%), validation (20%), and testing (10%). The dataset images were annotated by using an open-source tool called LabelImg. The bounding box coordinates were stored in XML format, converted into CSV, and finally, to the TF records [16]. The common/scientific names of each class along with the number of images (without augmentation) are shown in Table 2. An example of each annotated healthy and disease class is presented in Fig. 4.

5) DATA AUGMENTATION TECHNIQUES

When collecting the images for the dataset, some of them were taken in a group; cropping of those images increased the size of the dataset. Furthermore, several data augmentation

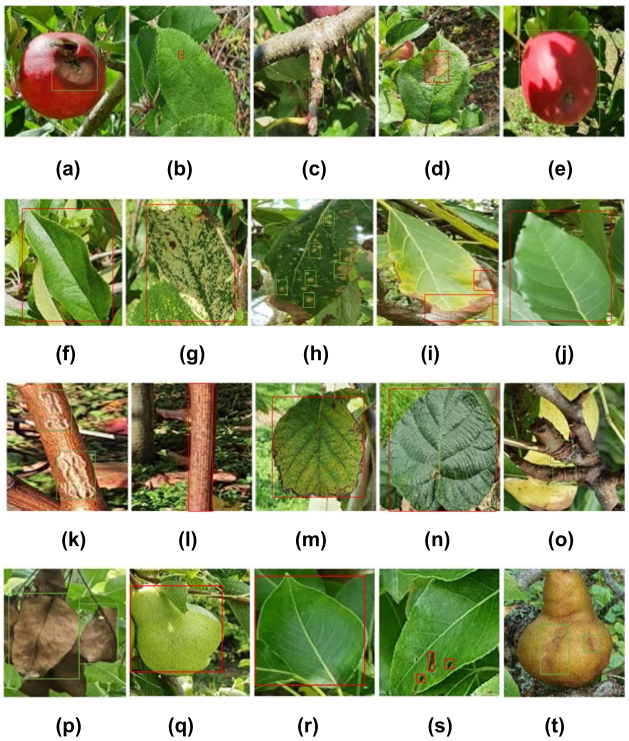


FIGURE 4. Annotated sample of each healthy and disease classes from the NZDLPlantDisease-v1 dataset: (a) Apple black rot, (b) Apple black spot (scab), (c) Apple European canker, (d) Apple glomerella leaf spot, (e) Apple healthy (fruit), (f) Apple healthy (leaf), (g) Apple mosaic virus, (h) Avocado algal leaf spot, (i) Avocado branch canker, (j) Avocado healthy, (k) Grapevine black spot, (l) Grapevine healthy, (m) Kiwifruit bacterial canker, (n) Kiwifruit healthy, (o) Pear canker, (p) Pear fire blight, (q) Pear healthy (fruit), (r) Pear healthy (leaf), (s) Pear scab, (t) Pear stony pit.

techniques were applied, such as a 30% increase and decrease in brightness, contrast, and sharpness [25]. Moreover, two noises are injected into the training images to study their effects and increase the variability in the dataset. In this regard, Gaussian and Laplacian noise were added by using the online available software named XnViewMP. The random intensity of 2.0 and 10.0 at a maximum scale of 10.0 and 50.0 was set, respectively. In addition, rotational/translational changes were also considered, including 90°, -90°, 180°, horizontal, and vertical changes. An example of augmented images for a kiwifruit bacterial canker is shown in Fig. 5.

The data augmentation techniques are grouped into five categories to thoroughly understand their effects on the performance of the DL model. These categories are only original (OO), original and change in translation/rotation (OT), original and color change (OC) (brightness, contrast, and sharpness), original with an injection of noise (Gaussian and Laplacian) and color change simultaneously (OCN), and finally a combination of all (OTCN).

C. DEEP LEARNING FRAMEWORK, HARDWARE SPECIFICATIONS, AND PERFORMANCE METRICS

All experiments are performed using the TensorFlow object-detection API. The DL models are trained using the transfer

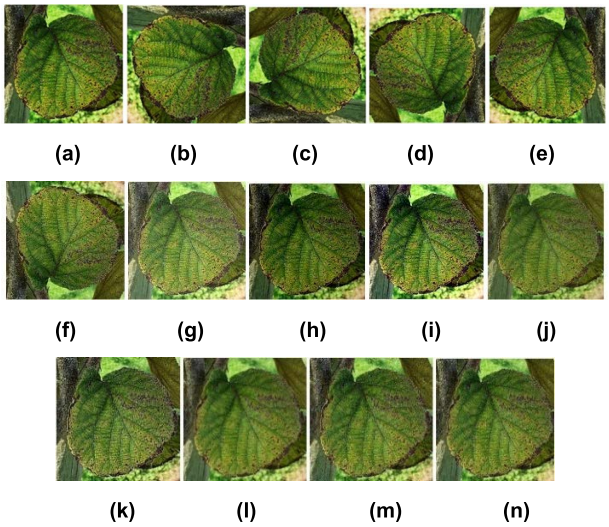


FIGURE 5. Samples of kiwifruit bacterial canker with/without data augmentation methods: (a) original, (b) 90°, (c) -90°, (d) 180°, (e) horizontal, (f) vertical, (g) high brightness, (h) brightness low, (i) contrast high, (j) low contrast, (k) sharpness high, (l) sharpness low, (m) Gaussian noise, (n) Laplacian noise.

learning technique with pre-trained weights on the COCO dataset. An NVIDIA GeForce GTX 1080 Ti graphics processing unit (GPU) is used with the following specifications: 11 GB memory, 1582 MHz boost clock, 3584 CUDA cores, and 484 GB/s memory bandwidth. The CuDNN library is imported to accelerate training.

The performance of the DL models is evaluated through the training and validation profiles in terms of various classification and localization losses. This helped to gain insight into the models by box classifier loss, region proposal network (RPN) loss, and total loss. Furthermore, the testing performance is presented using the mean average precision (mAP), which is a commonly used performance metric for object detection tasks [16].

D. DEEP LEARNING MODELS

Numerous well-known DL meta-architectures have been trained, tested, and compared on the generated dataset. These models include the Single Shot Multibox Detector (SSD) [33], Faster Region-based Convolutional Neural Network (RCNN) [34], Region-based Fully Convolutional Network (RFCN) [35], RetinaNet [36], and EfficientDet [37].

The complexity of the DL models is presented by training and detection time, architectural differences, and the number of parameters, as shown in Table 3.

E. ARCHITECTURAL OPTIMIZATION OF THE RFCN MODEL

1) FUNDAMENTALS OF RFCN

Following the proposed methodology, the RFCN is selected as the best DL model for the detection of plant diseases. The main idea of this DL model is to address the ambiguity between translational invariance (identifying a particular object at different pixel values) and translational variance (identifying the exact location of the object) using position-sensitive score maps. An RFCN is a two-stage

TABLE 3. Comparison of architectural features, number of parameters and training specification of DL architectures

DL architectures		Architectural features and loss equations	Number of parameters (in millions)	Training time (h)	Detection time (s/image)
Faster RCNN	ResNet-50	Removes the selective search (SS) method that was present in its previous models RCNN (which runs SS multiple times) [38] and Fast RCNN (which runs SS only once) [39]. Faster RCNN used Region Proposal Network (RPN) which reduced the number of runs significantly. $L[\{p_i\},\{t_i\}] = \frac{1}{N_c}\sum_i L_c(p_i, p_i^*) + \lambda\frac{1}{N_r}\sum_i p_i^* L_r(t_i, t_i^*)$ (1)	43	9.5	0.35
	ResNet-101		62	10.5	0.34
	Inception-v2		13	9	0.31
	Inception ResNet-v2		59	30	0.61
SSD	Inception-v2	A single-stage DL model performs the classification and localization task in a single forward pass. The bounding box regression technique was applied $L(x,c,l,g) = \frac{1}{N}[L_c(x,c) + \alpha L_L(x,l,g)]$ (2)	13	11.5	0.27
	MobileNet-v2		6	5.5	0.27
EfficientDet EfficientNet		A weighted bi-directional feature pyramid network (BiFPN) and a compound scaling method were proposed to allow an easy and fast multi-scale feature fusion, and scale of the depth, width, and resolution uniformly.	3.9	12	4.2
RFCN ResNet-101		Generation of position-sensitive score maps. Unlike RCNN-based models, the fully connected layer (FC) was removed after the region of interest (ROI). Therefore, the main complexity was relocated before ROI. $L_c = -\log(s_c^*)$ (3) $L(s, t_{x,y,w,h}) = L_c(s_c^*) + \lambda[c^* > 0]L_r(t, t^*)$ (4)	64	10.5	0.33
RetinaNet		Introduced feature pyramid network (FPN) to detect small objects in an image and focal losses to address the class imbalance problem. $FL(p_t) = -\alpha_t(1-p_t)^\gamma \log(p_t)$ (5)	36	14.5	2.1

L: overall loss, pi: the output score from classification branch for anchor i, t_i: output prediction of the regressor layer, N_c: number of anchors in mini-batch for classification, L_c: classification loss, p_i^*: ground truth label (0/1), λ: balancing parameter (for Faster RCNN), N_r: number of anchors in mini-batch for classification, L_r: regression loss t_i: output prediction of the regressor layer, N: , L_c: confidence loss, α: weight term, L_L: localization loss, $L_c(s_c^*)$: cross-entropy loss, λ: balancing parameter (for Faster RCNN), $[c^*>0]$ is an indicator, which is equal to 1 if the argument is true and 0 otherwise, c^*: ground-truth label of the region of interest (ROI), t^*: ROI's ground truth box, FL: focal loss, p: model's estimated probability for the class (RetinaNet), $\gamma \in [0, 5]$.

DL architecture. First, the input image is applied to the feature extraction layer using a convolutional neural network (CNN) to generate feature maps. These maps are applied to a convolutional layer to generate region-of-interest (ROI) proposals. In the former DL model named Faster RCNN, ROI proposals were used to extract the feature region in the feature map and to extract features to differentiate a particular class. However, in the RFCN, another convolutional layer is used to generate position-sensitive score maps. This map splits the ROI into k × k bins, where each bin is used to vote for the class to which the object belongs. Therefore, the main idea was to consider the characteristics of an object divided into a region k × k instead of as a whole. Both the RFCN and Faster R-CNN models have the same extraction of ROI proposals, but technical and computational differences arise in the application of a fully connected layer (FC) in each ROI proposal for Faster RCNN. In contrast, the RFCN generates only the proposed score maps, and the ROI is only used to vote for the regions in the score maps. Hence, the overall training and testing times of the RFCN network are much reduced than those of the Faster RCNN model [35]. This difference can also be observed in Fig. 6.

For w x h rectangular ROI, each bin is of size $\frac{w}{k} x \frac{h}{k}$. The x and y coordinates for (i,j)th bin for one of the slices of anchor

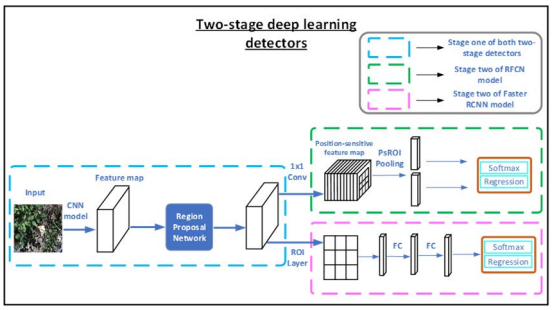


FIGURE 6. Architectural details of RFCN and faster RCNN models.

box bin are obtained by the following formula.

$$[i\frac{w}{k}] \le x < (i+1)\frac{w}{k}] \; and \; [j\frac{h}{k}] \le y < (j+1)\frac{h}{k}] \quad (6)$$

Then, the pooled response $r_c(i,j)$ on (i,j) bin for class c is equal to the sum of all the pixels within that bin coming from the position sensitive score maps. This sum is divided by number of pixels (n) as the layer before the softmax function is the average pooling layer. Finally, we take a vote by averaging them out or taking the maximum and get the position sensitive scores that lead to the softmax to predict

the class

$$r_c(i,j|\theta) = \sum_{(x,y)\varepsilon bin(i,j)} \mathbf{z_{i,j,c}}(x + \mathbf{x_0}, y + \mathbf{y_0}|)/\mathbf{n} \quad (7)$$

$$r_c(\theta) = \sum_{\mathbf{i,j}} \mathbf{r_c(i,j}|\theta) \quad (8)$$

where θ presents all learnable parameters, $\mathbf{z}_{i,j,c}$ is the one score map out of $k^2(C+1)$ score maps, and (x_0,y_o) is the top left corner of the ROI.

2) EMPIRICAL OBSERVATIONS ON THE POSITION-SENSITIVE SCORE MAPS

The default architectural settings of the RFCN attained satisfactory outcomes for most of the plant disease classes. However, the pear scab could not be detected, showed false-positive identification. It was confused with the apple black spot. This was due to the similarity in the symptoms of both diseases. These results motivated us to empirically investigate the main novelty of the RFCN model. In this regard, the spatial bin width and height for position-sensitive score maps are tuned/analyzed for this research. The main purpose of this step is to improve the average precision (AP) of the pear scab and maintain the high AP of other classes.

3) PERFORMANCE ENHANCEMENT THROUGH MODIFIED ANCHOR BOXES

The final step is the improvement of classes that achieved low AP (less than 80%). These classes include apple black rot, apple black spot, apple European canker, and pear healthy (leaves) classes. In this regard, this study explored the enhancement of anchor boxes in two steps: adjustment of scale sizes and aspect ratios. Here, the scale size is gradually modified, whereas the aspect ratios are reduced/enhanced in both a step-by-step (1:2, 1:3, 1:4, and so on) and reciprocal fashion (1:2, 2:1; 1:3, 3:1; etc.). The final output attained a high AP for all healthy and diseased classes. The following steps are taken to obtain a modified or enhanced version of the anchor boxes.

- First, the RFCN model is trained by the default specifications of the scale size and aspect ratio. Therefore, the scale width and height of 256 x 256 of combinations like 1:4, 1:2, 1:1, 2:1 with aspect ratio of 1:2, 1:1, 2:1 and 1:1 are considered.
- Afterward, smaller/larger scale sizes are added to the default to understand their effects on model performance.
- After obtaining the best combination of anchor box scales, the aspect ratios are added and enhanced to obtain further refinement in the detection of plant diseases. From the default aspect ratios, the reciprocal ratios such as 1:3, 3:1, and 1:4, 4:1 are applied.
- Subsequently, an empirical adjustment of the aspect ratio is performed, and a gradual reduction/enhancement of the aspect ratio is proposed to improve the AP of several classes. The combined effect of reciprocal and gradual changes in the aspect ratio is also studied.
- Finally, the training, validation profiles and testing outcomes are compared between the proposed modifications and the default settings.

F. IMAGE RESIZERS AND INTERPOLATORS

After obtaining the best combination of the data augmentation technique and DL architecture, the effects of image resizers on the model performance are studied. Aspect ratio and fixed shape resizers are used along with four types of image interpolators: bilinear, bicubic, area, and nearest neighbor.

G. WEIGHTS INITIALIZERS

Three weight initialization techniques are compared to optimize the performance of the best-suited DL architecture. By default, a truncated normal is used to remove dead neurons caused by the ReLU. Then, variance scaling is applied, which is beneficial to balance the variance of the output with the input layers [40]. The last initializer is a random normal initializer used to create tensors through a normal distribution.

H. BATCH NORMALIZATION

To accelerate training speed, batch normalization (BN) is used in this research. This technique solves the problem of internal covariate shift due to the variation in the input of the distribution of the neural network with the variation in the parameters of the previous layer [41].

The mini batch mean (μ_ϕ) for a mini-batch (ϕ), mini-batch variance (σ_ϕ^2) and normalize (affine transform) are evaluated for each row of input matrix (x_i) by:

$$\mu_\phi \leftarrow \frac{1}{N}\sum_{i=1}^{N} \mathrm{x_i} \quad (9)$$

$$\sigma_\phi^2 \leftarrow \frac{1}{N}\sum_{i=1}^{N} (\mathrm{x_i} - \mu\phi)^2 \quad (10)$$

$$\hat{x_i} \leftarrow \frac{\mathrm{x_i} - \mu_\phi}{\sqrt{\sigma_\phi^2 + \varepsilon}} \quad (11)$$

where N is the number of instances in mini batch, ε is added for the numerical stability.

There is a zero mean and variance for each component of x_i though the hidden units should have different distributions. Therefore, the normalization scheme learns the distribution by scaling the normalized values through scaling (γ) and shifting (β) parameters and evaluated the output of the batch normalization as follows:

$$y_i \leftarrow \gamma.\hat{x_i} + \beta \equiv B\mathrm{N}_{\gamma,\beta}(\mathrm{x_i}) \quad (12)$$

I. DEEP LEARNING OPTIMIZERS AND SELECTION OF HYPERPARAMETERS

In this study, three DL optimizers are used. Stochastic gradient descent (SGD) with momentum [42] is applied as a default optimization algorithm, later, root mean square propagation (RMSProp) [43] and adaptive moment estimation (Adam) Adam [44] are used to optimize the weights. A brief overview of the DL optimizers is given as follows:

1) SGD WITH MOMENTUM

The stochastic gradient descent (SGD) is the most commonly used optimization algorithm for neural networks. The momentum version of the SGD has a great capability of faster convergence compared to the original SGD optimizer. The exponentially weighted averages (*Vdw* and *Vdb*) are used to evaluate the gradient and use the gradient to update the weights (W) and biases (b). The algorithm uses the following equations:

$$\mathrm{Vdw_t} = \beta * \mathrm{Vdw_{t-1}} + (1-\beta) * \mathrm{dw_t} \quad (13)$$

$$\mathrm{Vdb_t} = \beta * \mathrm{Vdb_{t-1}} + (1-\beta) * \mathrm{db_t} \quad (14)$$

$$\mathrm{W_t} = \mathrm{W_{t-1}} - \mathrm{lr} * \mathrm{Vdw_t} \quad (15)$$

$$\mathrm{b_t} = \mathrm{b_{t-1}} - \mathrm{lr} * \mathrm{Vdb_t} \quad (16)$$

where, β, *lr*, *dw,* and *db* present momentum, learning rate, gradients of the weights, and biases, respectively.

2) RMSProp

This DL optimizer allows to select a large learning rate. It works on the idea of using the moving average of the squared gradient and dividing the gradient by square root the mean square, using the following equations:

$$\mathrm{Vdw_t} = \beta * \mathrm{Vdw_{t-1}} + (1-\beta) * \mathrm{dw_t^2} \quad (17)$$

$$\mathrm{Vdb_t} = \beta * \mathrm{Vdb_{t-1}} + (1-\beta) * \mathrm{db_t^2} \quad (18)$$

$$\mathrm{W_t} = \mathrm{W_{t-1}} - \mathrm{lr} * \frac{\mathrm{dw_t}}{\sqrt{\mathrm{Vdw_t}} + \varepsilon} \quad (19)$$

$$\mathrm{b_t} = \mathrm{b_{t-1}} - \mathrm{lr} * \frac{\mathrm{db_t}}{\sqrt{\mathrm{Vdb_t}} + \varepsilon} \quad (20)$$

where, ε is used for the numerical stability in the denominator.

3) ADAM

The Adam optimizer is a combination of RMSProp and SGD with momentum optimizers. Like RMSProp, Adam takes squared gradients to scale the learning rate and uses moving average of the gradients similar to the SGD with momentum. As it has an adaptive learning rate, it calculates separate learning rates for each parameter. Adam contains estimations of first (mean) and second moments (uncentered variance) of gradient that are used to adapt the learning rate for each weight of the DL model/neural network. Whereas the moment is considered as the expected value of a variable to the power of n. The first (*mdw, mdb*) and second moment *(vdw, vdb)* estimates are evaluated by equations (21)-(24).

$$\mathrm{mdw_t} = \beta_1 * \mathrm{mdw_{t-1}} + (1-\beta_1) * \mathrm{dw_t} \quad (21)$$

$$\mathrm{mdb_t} = \beta_1 * \mathrm{mdb_{t-1}} + (1-\beta_1) * \mathrm{db_t} \quad (22)$$

$$\mathrm{vdw_t} = \beta_2 * \mathrm{vdw_{t-1}} + (1-\beta_2) * \mathrm{dw_t^2} \quad (23)$$

$$\mathrm{vdb_t} = \beta_2 * \mathrm{vdb_{t-1}} + (1-\beta_2) * \mathrm{db_t^2} \quad (24)$$

The bias-corrected first ($\hat{mdw}$ $\hat{mdb}$) and second ($\hat{vdw}$, $\hat{vdb}$) moment estimates are evaluated by equations (20)-(23).

$$\hat{\mathrm{mdw}}\,\mathrm{t} = \mathrm{mdw_t}/(1-\beta_1^{\mathrm{t}}) \quad (25)$$

$$\hat{\mathrm{mdb}}\,\mathrm{t} = \mathrm{mdb_t}/(1-\beta_1^{\mathrm{t}}) \quad (26)$$

$$\hat{\mathrm{vdw}}\,\mathrm{t} = \mathrm{vdw_t}/(1-\beta_2^{\mathrm{t}}) \quad (27)$$

$$\hat{\mathrm{vdb}}\,\mathrm{t} = \mathrm{vdb_t}/(1-\beta_2^{\mathrm{t}}) \quad (28)$$

Then, the weights and biases are evaluated by:

$$\mathrm{W_t} = \mathrm{W_{t-1}} - \mathrm{lr} * \frac{\mathrm{mdw^t}}{\sqrt{\mathrm{vdw^t}} + \varepsilon} \quad (29)$$

$$\mathrm{b_t} = \mathrm{b_{t-1}} - \mathrm{lr} * \frac{\mathrm{mdb^t}}{\sqrt{\mathrm{vdb^t}} + \varepsilon} \quad (30)$$

where ε is equal to 10^{-8}

4) SELECTION OF HYPERPARAMETERS

The hyperparameter values are selected using the random search method [45], presented in Table 4. For example, the learning rate (*lr*) of the SGD optimizer to train the RFCN model was tuned exponentially from 10^{-5} to 10^{-1}, while the momentum (*mom*) was tuned with a difference of 0.1. The hyperparameter tuning was started with *lr* of 10^{-1} and zero mom, the RFCN did not get the training convergence. Then, the *lr* was started to reduce and mom was increased. The training of the RFCN model started to settle down. For example, at *lr* of 10-3 and *mom* of 0.8, the mAP was 61.60% with a total training loss of around 0.41%. A further reduction in the *lr* positively influenced the performance of the RFCN. At the learning rate of 10^{-4} and mom of 0.8, the training loss was reduced to 0.23% with the mAP of 73.256%. But a further increase in *lr* (10^{-5}) significantly increased the training time. Therefore, small random changes were made for *lr* and *mom* and the performance of the RFCN model was checked in various values. It was found that *lr* of in 3×10^{-4} and *mom* of 0.9, the loss was reduced to around 0.09% and the mAP improved significantly to 74.47%.

TABLE 4. Hyperparameters of deep learning optimizers.

DL optimizers	DL meta-architectures	Hyperparameters
SGD with momentum	Faster RCNN ResNet-50	lr = 2 x 10^{-4}, mom = 0.9
	Faster RCNN Inception-v2	lr = 1 x 10^{-4}, mom = 0.9
	EfficientDet	lr = 3 x 10^{-4}, mom = 0.9
	SSD Inception-v2	lr = 3 x 10^{-4}, mom = 0.9
	SSD MobileNet-v2	lr = 2 x 10^{-4}, mom = 0.9
	SSD ResNet-50 (RetinaNet)	lr = 4 x 10^{-4}, mom = 0.9
	Faster RCNN Inception ResNet-v2	lr = 2 x 10^{-3}, mom = 0.9
SGD with momentum	Faster RCNN ResNet-101	lr = 3 x 10^{-4}, mom = 0.9
RMSProp		lr = 2.5 x 10^{-4}, rho = 0.9, mom = 0.9, eps = 1 x 10^{-3}
Adam		lr = 1 x 10^{-4}, eps = 1.0^{-2}
SGD with momentum	RFCN ResNet-101	lr = 3 x 10^{-4}, mom = 0.9
RMSProp		lr = 3 x 10^{-4}, rho = 0.9, mom = 0.9, eps = 1 x 10^{-3}
Adam		lr = 2 x 10^{-4}, eps = 1.0^{-3}

lr: learning rate, *mom*: momentum, *eps*: epsilon

J. VALIDATION METHODS

1) STRATIFIED FIVE-FOLD CROSS-VALIDATION

The proposed DL-based approach has been validated using two techniques. First, owing to the class imbalance problem due to the different number of images of each class (can be seen in Table 2), a stratified cross-validation method is used. This method retains the particular number of data points/sample size of each class in each fold [46]. It ensures the unbiased distribution of the dataset among all folds. Otherwise, random sampling could generate bias in the folds when all dataset images are randomly shuffled and split into a certain number of folds.

2) TESTING ON AN EXTERNAL DATASET

Another contribution of this study is the validation of the final results using an external test dataset (obtained by a random search on various websites). This was done to show the effectiveness and robustness of the work that presented DL-based approach would also be applicable under different environmental conditions than the one used for dataset generation.

III. RESULTS AND DISCUSSIONS

After the dataset generation, the proposed approach is divided into several steps to get the optimized DL model for plant disease detection. The results presented in this section follow the methodology of the research, as shown in Fig. 1. First, a comparison of the DL architectures was performed. This step was performed to obtain the top two models. The training and validation plots are presented (to understand the performance of several DL models) and the detection results (to evaluate the mAP). The best two DL models were trained with all data augmentation methods to understand their effects. Later, the effects of image resizing techniques and interpolators are provided in terms of training, validation losses and mAP. These methods evaluated the impact of the input image on the DL model. Afterward, performance optimization has been explained by weight initializers, batch normalization, and DL optimizers. The effects of various parameters of the weight initializers are also provided. Similarly, the performance of the best-obtained DL model was evaluated in the presence of batch normalization, to show the better convergence ability of the DL model. DL optimizers are also compared to optimize the weights of the best-obtained DL architecture.

After the optimization of the DL model, further in-depth class-wise analysis has been performed. In this regard, the performance of the individual classes is evaluated. The classes that attained the lowest AP were explicitly focused. This step also aimed to maintain the high AP of the other classes obtained in the previous step. The position-sensitive score maps are analyzed, as it was one of the major novelties of the RFCN model (the best-obtained model). The detection results are shown to understand the impacts of the spatial bin width and heights of the score maps. Furthermore, the anchor boxes were enhanced to show the influence of various anchor box scales and aspect ratios. The results are shown by the training and validation plots and average precision of each class, along with the mAP of each enhanced version. Finally, the stratified k-fold cross-validation method was used due to the class imbalance problem in the proposed dataset and to validate the final mAP of the optimized DL model.

A. COMPARISON BETWEEN DL ARCHITECTURES

First, the DL models are trained on the original (without augmentation) images. Subsequently, the two best models are retrained on the augmented images. It is empirically found that DL architectures should be trained to 200K steps to achieve training convergence. The input images are resized to 300 $\times$ 300 pixels with fixed image resizers for SSD MobileNet-v2 and SSD Inception-v2 and 640 $\times$ 640 pixels for SSD ResNet-50 (RetinaNet). An aspect ratio resizer with minimum and maximum pixel dimensions of 600 and 1000, respectively, is considered for the models including all versions of Faster R-CNN and RFCN. The EfficientDet model is also trained with an aspect ratio resizer with 512 minimum and maximum pixels, according to the GPU requirement. Furthermore, SGD with a momentum optimizer is used to train the models for this stage of the research. Different batch sizes are tested, and the most reasonable is found to be 4 to reduce the trade-off between accuracy and training time. Four models required the lowest iteration steps of 170K to achieve training convergence: Faster RCNN ResNet-50, RFCN ResNet-101, EfficientDet, and RetinaNet. The lowest training times are obtained for SSD MobileNet around 5.5 h. However, the Faster RCNN Inception ResNet-v2 required the highest time to complete the training. The following observations are made on the training and testing performance of the DL models.

1) TRAINING PERFORMANCE

- Plots of the total training and validation losses for each model are shown in Fig. 7. It can be observed that the Faster R-CNN ResNet-101 and RFCN ResNet-101 models attained the lowest total training and validation losses of approximately 0.05-0.08%, 0.06-0.09%, and 0.04-0.2%, 0.06-0.18% respectively. Both models took around 10.5 hours to achieve convergence.
- However, the versions of SSD models with Inception-v2 and MobileNet-v2 have approximately 1.5% total loss. This is comparatively higher than that of the other DL models, apparently reflected in their detection results as a low mAP as shown in Table 5.
- Later, the two best models are retrained with augmented images due to their lowest training and validation losses (after training on non-augmented images), including RFCN ResNet-101 and Faster RCNN ResNet-101. Their loss plots are shown in Fig. 8. It can be concluded from the plots that RFCN ResNet-101 has a slightly lower training and validation losses of approximately 0.7% and 1.0%, respectively.

TABLE 5. Summary of the plant disease detection results by the DL architectures in terms of average precision (in %) of each class divided into various augmentation techniques.

Healthy and disease classes	DL architectures with backbone models																
	Faster RCNN		EfficientDet	SSD			Faster RCNN						RFCN				
	ResNet-50	Inception-v2	EfficientNet	Inception-v2	MobileNet-v2	ResNet-50 (RetinaNet)	Inception ResNet-v2	ResNet-101					ResNet-101				
	OO	OO	OO	OO	OO	OO	OO	OO	OC	OT	OCN	OTCN	OO	OC	OT	OCN	OTCN
A_blk_rot_f	62.77	64.28	0	18.18	33.33	0	63.73	64.5	74.33	80.66	80.1	72.61	65.41	76.72	71.77	39.63	77.17
A_blk_spot	3.54	2.14	0	1.44	4.95	0	10.82	11.31	23.69	9.01	0	0	16.15	5	5.18	5	8
A_e_canker	81.57	39.58	49.09	0	18.18	49.09	67.8	33.5	33.37	51.55	85.34	83.4	40.27	80.19	68.83	6.01	95.78
A_gl_lf_spot	73.91	82.27	44.55	41.67	51.37	0	78	82.03	72.43	79.67	87.7	87.99	91.98	92.86	92.24	62.08	97.12
A_healthy_f	94.07	95	42.83	90.91	100	0	90.55	92.75	100	100	90.24	80.08	100	95.91	99.92	100	88.88
A_healthy_l	49.21	61.6	40	63.07	43.15	49.09	80.01	62.55	47.18	74.65	80.99	90.24	63.07	86.82	65.34	56.53	99.97
A_m_virus	28.04	33.19	43.9	32.78	32.91	47.27	38.74	42.18	58.41	45.11	7.57	7.25	40.97	31.07	65.77	33.54	9.19
Av_alg_lf_spot	70.6	70.32	0	90.26	88.11	0	58.15	77.62	26.55	71.57	12.51	5.23	75.62	9.55	63.93	10.88	8.26
Av_br_canker	53.59	70.67	0	32.66	22.98	0	77.69	55.88	77.34	80.92	19.31	6.31	76.21	54.48	99.8	46.5	8.94
Av_healthy_l	65.41	75.17	44.55	43.54	32.07	44.55	71.91	87.46	90.38	100	0.05	0	79.47	5.19	100	57.7	8
G_blk_spot_c	76.81	76.82	0	42.46	0	0	61.36	56.55	29.97	53.12	3.28	1.04	71.27	15.85	88.39	33.26	8
G_healthy_c	94.43	92.35	0	90.91	81.82	0	89.32	100	100	100	99.88	67.84	100	95.91	99.84	95.91	63.63
Kf_bac_canker	95	95	47.74	100	64.69	56.56	91	92.91	76.23	100	90.73	69.66	99.52	95.91	100	99.78	57.44
Kf_healthy_l	95	95	65.04	100	100	50	91	100	89.24	100	72.74	7.14	100	85.16	100	95.91	8.58
P_canker	87.23	78.23	40	36.36	61.74	58.18	86.8	81.59	82.91	89.94	42.87	7.54	82.53	55.76	97.56	99.76	8
P_fr_blight	34.45	57.91	0	36.36	36.36	0	53.34	62.83	53.12	43.28	53.69	43.11	81.08	76.23	24.16	100	47.79
P_healthy_f	95	95	0.83	100	100	0	91	100	100	92.91	100	90.91	100	100	95.91	100	98.91
P_healthy_l	95	95	67.27	100	100	58.18	91	100	23.76	48.54	99.96	88.97	100	99.68	24.37	24.55	81.83
P_s_pit	66.7	61.52	0	27.27	57.77	0	87.38	74.73	97.73	82.76	78.67	99.01	50.6	94.76	98.85	67.49	98.19
P_scab	45.88	49.79	58.18	26.26	26.33	49.09	49.96	66.99	0.29	59.62	56.72	1.85	55.34	45.65	73.26	67.22	13.78
mAP (%)	68.41	69.54	27.20	53.71	52.79	23.10	71.48	72.27	62.85	73.16	58.12	45.51	74.47	65.13	76.76	60.09	49.37

- It can be noticed from Figs. 7-8 that the DL models did not suffer from overfitting. Both training and validation losses were settled down with a small fluctuation and there was no such sudden increase in the validation loss after achieving the convergence in both the absence and presence of the augmented images.

2) TESTING PERFORMANCE

- The results obtained from the test dataset (without augmentation) revealed that the RFCN ResNet-101 model achieved the highest mAP of 74.47%, followed by Faster RCNN ResNet-101 and Faster RCNN Inception ResNet-v2, as shown in Table 5. This is because RFCN achieved a high AP of 10 healthy/disease classes. A sample of each class is shown in Fig. 9 (a).
- The Faster RCNN trained with Inception ResNet-v2 and ResNet-101 attained a higher mAP than the rest of the models, including Faster RCNN ResNet-50, Faster RCNN Inception-v2, SSD models, RetinaNet and EfficientDet. The Faster R-CNN ResNet-101 is found to be the most useful model for the healthy class of avocado (Av_healthy_l) and attained the highest AP among all DL models. It is also noticed that some of the testing images of apple glomerella leaf spot and pear fire blight obtained false positive and false negative detections, respectively, with Faster R-CNN ResNet-101, as shown in Fig. 9 (b). Similarly, classes such as apple healthy leaves and pear stony pits are well detected using Faster Inception ResNet-v2, as shown in Fig. 9 (c-d).
- Although the RFCN model attained the highest mAP, it misclassified some of the testing images of the classes, such as the stony pit on the pear, as shown in Fig. 9 (d).
- The testing performance of the models, including EfficientDet and RetinaNet, is unsatisfactory. This was due to several classes remaining undetected and giving false positive results, as presented in Fig. 9 (e).
- It can also be seen from Table 5 that the black spot on the apple leaves failed to be detected and localized by all DL models, an example from each DL model is presented in Fig. 9 (f). Some of the classes attained 0% average precision when trained by the models like EfficientDet, SSD Inception-v2, SSD MobileNet-v2, and SSD ResNet-50. Because these models failed to detect a few of the classes, that was observed in two ways. First, the testing images of those classes were undetected, second, the false positive results were obtained due to the

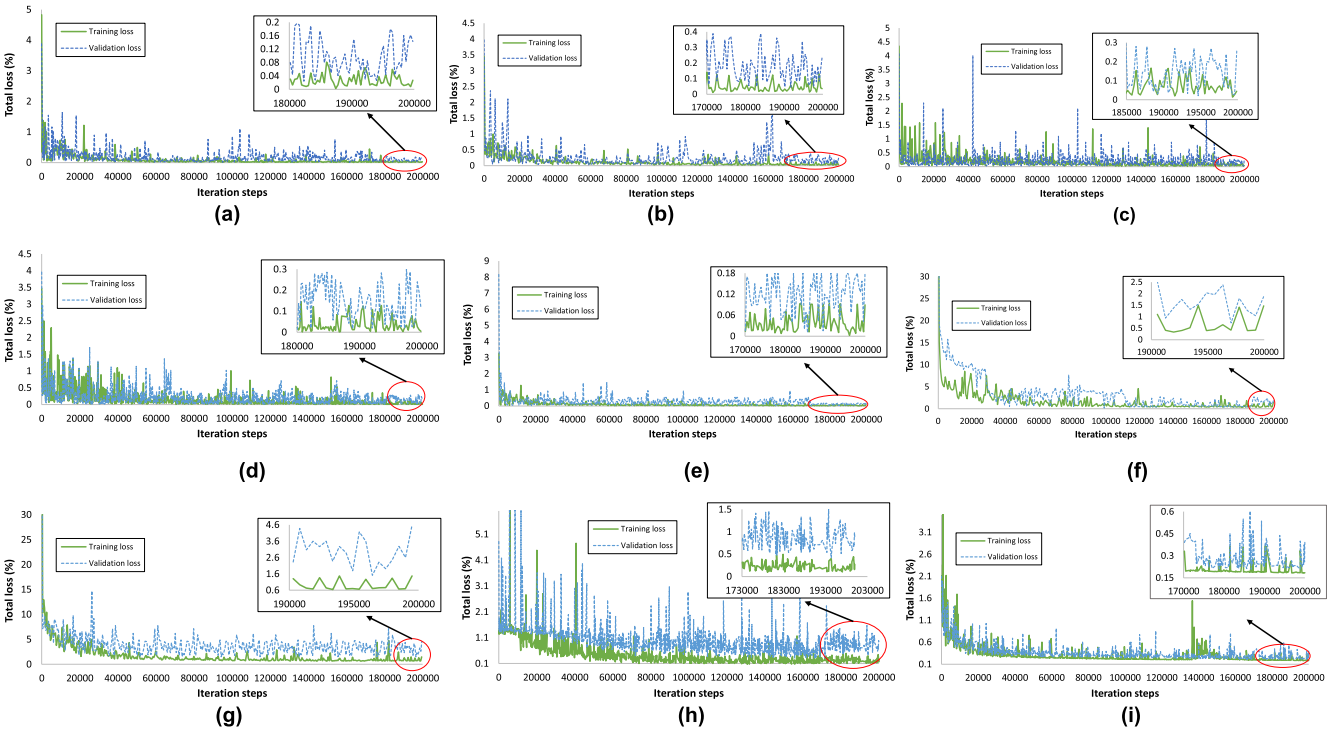


FIGURE 7. Total loss plot for deep learning models before augmentation: (a) Faster RCNN ResNet-101, (b) Faster RCNN ResNet-50, (c) Faster RCNN Inception-v2, (d) Faster RCNN Inception ResNet-v2, (e) RFCN ResNet-101, (f) SSD MobileNet-v2, (g) SSD Inception-v2, (h) EfficientDet, (i) RetinaNet.

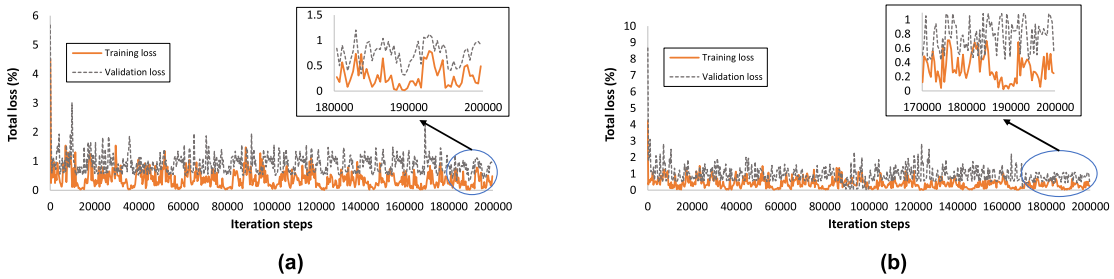


FIGURE 8. Total loss plot for deep learning models after the application of all augmentation techniques: (a) Faster RCNN ResNet-101, (b) RFCN.

confusion with another plant disease/healthy classes. In Fig. 9 (f), an example of the undetected/false negative outcome for apple black spot is presented for EfficientDet and RetinaNet.

- Further analysis was required to validate the selection of the best DL model for the next phase of research.
- In this regard, the top two DL models (RFCN ResNet-101 and Faster RCNN ResNet-101) in terms of the lowest training, validation losses and the highest mAP are retrained using augmented images (considering all 13 augmentation categories).
- RFCN ResNet-101 has achieved a higher mAP than Faster RCNN ResNet-101, as shown in Table 5.
- Some classes, including apple glomerrella leaf spot, apple European canker, and apple healthy classes, with the RFCN model, have achieved a higher AP compared to the Faster RCNN ResNet-101. A few examples from RFCN are shown in Fig. 10.
- Classes such as apple black rot, apple European canker, and healthy leaves of apples have improved their AP after training the RFCN through augmented images. On the contrary, the AP of almost 13 classes is significantly reduced by training in the augmented images, as shown in Table 5.
- The main finding of this step is that the RFCN model has achieved the highest mAP with and without augmented images. Another important observation is that the augmented images helped improve the AP of only a few classes. Moreover, the RFCN has shown its ability to address problems such as the detection of plant diseases in different organs and the identification of disease in different fruits, as shown in Figs. 9-10. However, the

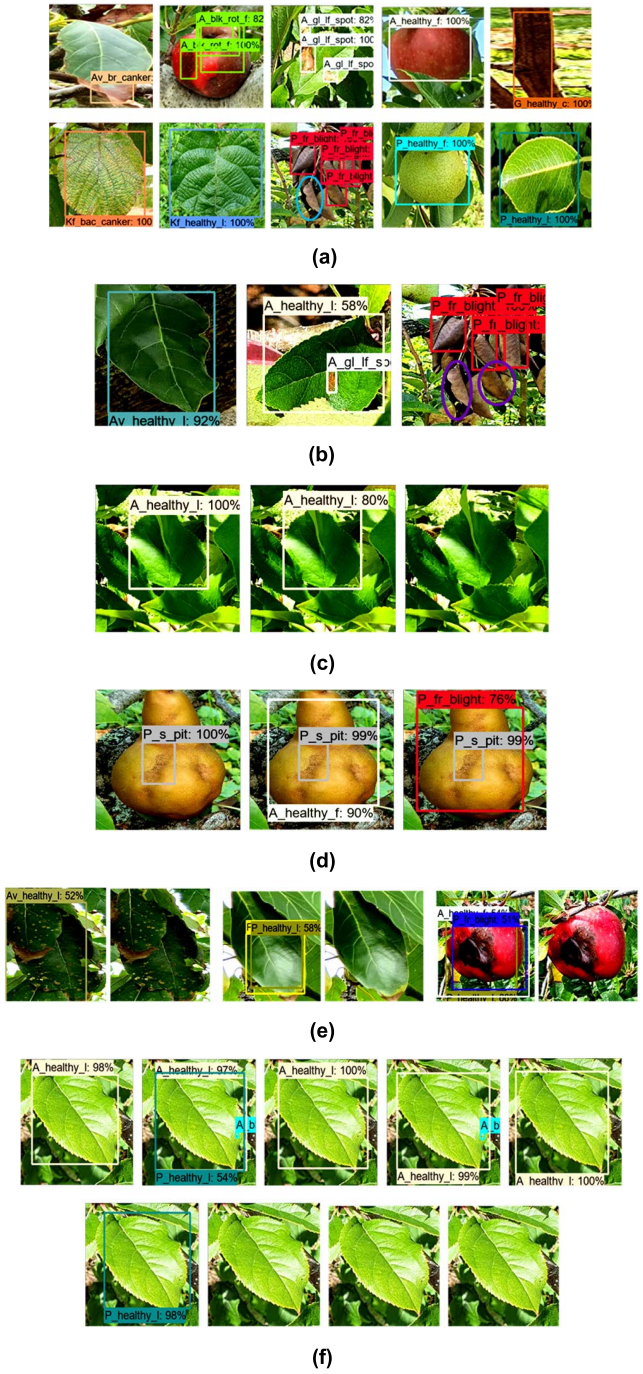


FIGURE 9. Detection results by various DL models trained without augmented images. (a) true positive and false negative (P_fr_blight) by RFCN, (b) true positive (Av_healthy_l), false positive (A_gl_lf_spot), and false negative results (P_fr_blight) with Faster RCNN ResNet-101, (c) true positive outcome for A_healthy_l class with Faster RCNN Inception ResNet-v2, Faster RCNN ResNet-101, and false negative for RFCN (first from the right), (d) P_s_pit with Faster RCNN Inception ResNet-v2, Faster ResNet-101, and RFCN (left to right), (e) false positive and false negative results with EfficientDet and RetinaNet, (f) false positive for A_blk_spot by RFCN, Faster RCNN ResNet-101, Faster RCNN Inception ResNet-v2, Faster RCNN ResNet-50, and Faster RCNN Inception-v2, (first row – left to right), and false positive, false negatives by SSD MobileNet-v2, SSD Inception-v2, EfficientDet, and RetinaNet (second row – left to right).

low AP of several classes was attained in the presence of augmentation techniques. This motivated us to individually evaluate the effects of the augmentation techniques. This helped to understand the reason for performance degradation after the application of 13 types of augmentation methods for this step of the study.

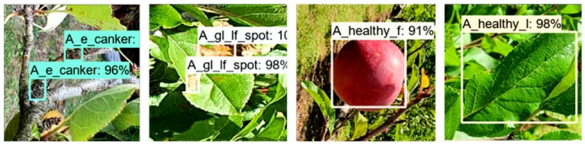


FIGURE 10. True positive results of RFCN after training on augmented images.

B. EFFECTS OF DATA AUGMENTATION TECHNIQUES

The effects of data augmentation methods are studied by dividing them into five categories as described in Section II-B (5). The two top DL models are trained on all five groups of augmented images, including RFCN ResNet-101 and Faster R-CNN ResNet-101. The important results from this phase of the study are discussed in the following.

- The RFCN model has achieved the highest mAP after training with the OT data augmentation category, followed by the results obtained through the OO images, as shown in Table 5. However, comparatively lower mAP values are observed for OC and OCN. Whereas, RFCN has achieved the lowest mAP with the OTCN.
- To further perform an in-depth analysis of the data augmentation techniques, a class-wise analysis is performed. For example, OT is found to be the best method due to its higher AP by the RFCN in eight classes of healthy individuals and diseases. Therefore, the OT category has attained superior results.
- The effectiveness of the OT category is also validated using Faster RCNN ResNet-101, as shown in Table 5. The mAP is higher in the OT group than in all other categories. However, with the OTCN, the model has achieved the lowest mAP value.
- There could be several reasons for performance degradation when training with OC and OCN. The nature of the real agricultural environment could contribute to the confusion in discriminating between plant diseases. Because the real field contains several background elements, a change in color or addition of noise to the original images distracts/fails the model to extract and learn the specific and distinct features of the disease symptoms. There may be similarities between the symptoms of the disease and background elements [47] after adding noise and changes in brightness, contrast, and sharpness. This has resulted in a low AP for the individual classes and a low mAP for the 20 classes.
- The above statement can be further understood by taking examples of some classes that achieved comparatively lower AP. For example, avocado branch canker is confused with apple glomerella leaf spot, healthy avocado

leaf with healthy pear leaves, and apple mosaic virus with healthy apple healthy leaves. Similarly, the black spot-on grapevine cane could not be detected and/or is misclassified as an apple European canker. The stony pit virus on the pear is also confused with fire blight. Furthermore, in avocado leaves, several algal leaf spots are not detected. Some examples of false-positives and undetected outcomes are shown in Fig. 11.

- In contrast, the improved performance with OT has shown the practical aspect of this work. Because the location of disease spots in a real agricultural environment varies from one plant to another. Furthermore, if a DL model such as RFCN can detect the disease in the translated and rotated images, it shows the importance of the OT-based data augmentation technique for the identification of plant diseases.
- Fig. 11 showed the significance of the original with translational changes (OT-based augmentation method) by taking examples of five plant classes. These classes consist of diseases in plant organs including leaves (apple and avocado), fruit (pear), and cane (grapevine), including avocado branch canker, avocado healthy leaves, apple mosaic virus, grapevine black spot, and pear stonypit. An improvement of 23.59, 20.53, 24.8, and 17.12%, and 48.25%, respectively, was attained as compared to the previous step.

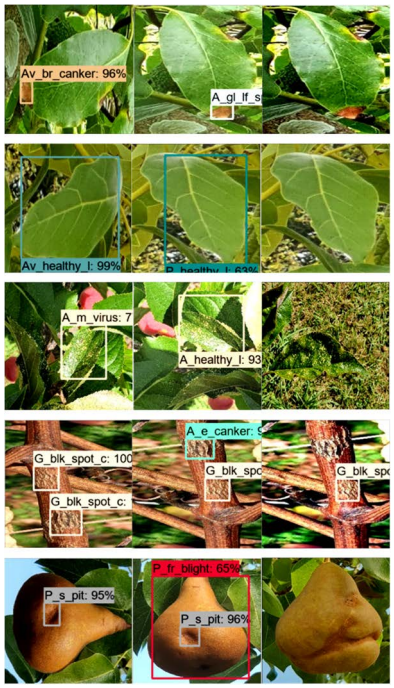


FIGURE 11. Detection outcomes of RFCN model by three augmentation categories. True positive, false positive, and false-negative results from left to right in each example belong to OT, OC, and OCN respectively.

C. EFFECTS OF IMAGE RESIZERS AND INTERPOLATORS

The next step is to study the effects of image resizers with different interpolators on RFCN. In this regard, the RFCN model is initially trained using an aspect ratio resizer and a bilinear interpolator. Eight possible combinations of image resizers and interpolation methods are used to obtain the best results, as shown in Table 6. The performance of each combination was not only evaluated by mAP, training, and validation losses, but also by the training time as considered in [48]. The main observations and discussions of this step of the analysis are presented below.

- The aspect ratio is considered the default resizing technique with minimum and maximum dimensions of 600 and 1000 pixels, respectively [35]. This image resizing technique is used in conjunction with bilinear interpolation.
- As presented in Table 6, the aspect ratio resizer with the other three interpolators, such as bicubic, area, and nearest neighbor, has degraded the performance of RFCN in terms of lower mAP.
- Later, a fixed-shape resizer with a default value of 300 × 300 pixels is applied with bilinear interpolation. Subsequently, three other interpolators are tested. The RFCN model trained with the bicubic interpolator with fixed shape resizer required the lowest training time, training loss (0.52%) and validation loss (0.85%) as shown in Table 6. Both losses were lower than the loss acquired with the default resizer/interpolator, as shown in Fig. 12.
- There is no such sign of overfitting of the RFCN model as the validation loss was also settled to the final value and no such increase in the loss was observed. It is validated with a higher mAP of 80.59%, which is 3.83% better than that obtained using the default settings. Furthermore, the AP of avocado algal leaf spot, pear fire blight, and healthy pear leaves are significantly improved to 88.55%, 98.16%, and 49.9%, respectively.
- The bilinear interpolator with a fixed-shape resizer has also performed slightly better than the nearest neighbor and area interpolation.
- Bicubic interpolation considers a 4 × 4 or 16-pixel square and evaluates the resulting interpolated pixels, compared to 2 × 2 pixels for bilinear interpolation. Therefore, better-quality images are obtained for the healthy and disease classes to be fed into the RFCN model, producing a better mAP.
- In conclusion, both the training and testing performances of the RFCN model are improved with an enhancement in the AP of the three classes (after training with a fixed-shape resizer along with a bicubic interpolator). In the future, other relevant datasets can be tested using the combinations of resizers and interpolators presented in this phase of the research.

D. EFFECTS OF WEIGHT INITIALIZERS, BATCH NORMALIZATION, AND DL OPTIMIZERS

The next phase of the proposed approach is the optimization of the RFCN model. The appropriate selection of weight

TABLE 6. Effects of image resizing techniques and interpolators on the RFCN ResNet-101 model.

Image resizing techniques/interpolators	Aspect ratio				Fixed Shape			
	Bilinear	Bicubic	Area	Nearest neighbor	Bilinear	Bicubic	Area	Nearest neighbor
Training loss (%)	0.60	0.625	0.62	0.61	0.55	0.52	0.56	0.56
Validation loss (%)	1.1	1.35	1.15	1.2	0.83	0.85	0.88	0.91
Training time (sec/steps)	0.189	0.189	0.195	0.195	0.141	0.138	0.147	0.147
mAP (%)	76.76	70.06	71.96	70.62	75.59	80.59	74.33	74.63

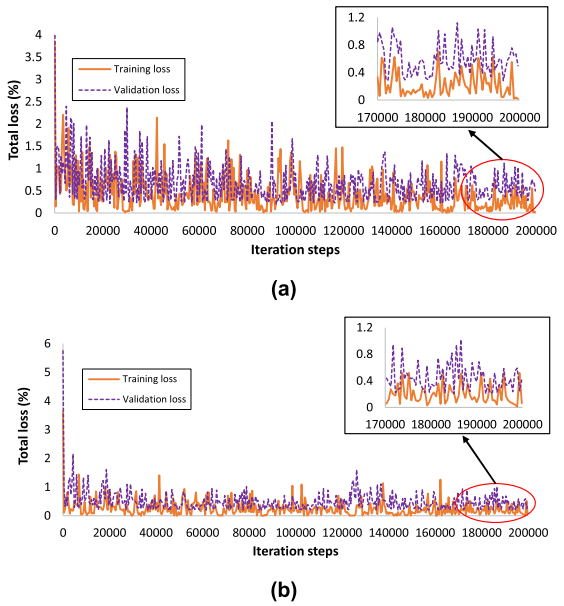


FIGURE 12. Total loss plots from two different image resizers/ interpolators: (a) aspect ratio resizer with bilinear interpolator, (b) fixed shape resizer with bicubic interpolator.

initializers along with their parameters solves the problems of vanishing and exploding gradient descent. Three weight initialization methods are used: truncated normal (default initializer), scaling variance, and random normal. After determining the most suitable initialization technique, the effects of batch normalization are studied. Then, the best DL optimizer is selected, and the hyperparameters are tuned using a random search method [45]. These steps are performed before proposing any modifications to the RFCN model. The training and testing evaluations of this phase of the study are summarized below.

- First, the RFCN model is trained using a truncated normal initializer. Subsequently, it is trained using the scaling variance and random normal initializers.
- The random normal initializer has achieved the lowest training loss and the highest mAP with a standard deviation of 0.01 and a mean value of zero. It has been observed that the selection of an optimum value for standard deviation and mean plays an important role in the performance of the model.
- When searching for the appropriate values of standard deviation and mean, the random normal initializer was initially used with a default standard deviation of 1 and a mean of zero. However, these values were unable to achieve convergence. Therefore, the standard deviation started to tune exponentially. It was empirically found that 0.01 was the most suitable standard deviation and the mean value of zero remained the same. This was due to the lowest training and validation losses, which resulted in a slight improvement of 0.916% in the mAP. Furthermore, the extraction of distinct features was also improved, as the AP of some of the classes was enhanced with the described settings. These classes include apple black rot, apple European canker, apple glomerella leaf spot, apple mosaic virus, and healthy pear (leaves), with an improvement of 8% to 28%. Similar standard deviations and mean values were also suitable for the truncated normal initializer.
- The best settings for scaling variance were single scaling factors with a uniform distribution and considering the average of the input and output units in the weight tensor (Fan_avg). However, these parameters did not contribute to improving the model performance and attained an mAP of only 70.20%.
- The random normal initializer has produced the best result for a particular application of the detection of plant disease by the RFCN model. Therefore, it can be concluded from its basic functionality that the initialized weights through the generation of tensors with normal distribution performed well for the selected problem. Moreover, theoretically, the random normal is supposed to work with weights initialized very close to zero. So, each neuron of the network does not perform the same calculation.
- It was experimentally found that a small standard deviation value was not suitable. For example, at a standard deviation of 0.001, the performance declined in terms of mAP to 78.33%, compared to mAP at 81.50%, obtained with a 0.01 standard deviation.
- The next step was the use of batch normalization. The RFCN is trained with the default values of epsilon and decay of 0.001 and 0.99, respectively. It is also noted that the training convergence is achieved earlier to around 160K steps from around 170K iterations. Therefore, it can be concluded that batch normalization reduced the overall training time and showed a fast convergence ability [49].
- The decay and epsilon were started to tune, and it was experimentally found that the lower value of decay and the higher value of epsilon improved the performance

of RFCN. Therefore, the decay was set to 0.5 and the epsilon was 0.01. The training performance was slightly improved to around 0.515%.

- The testing performance of the RFCN model was also significantly improved. The model attained an mAP of 85.94% with an improvement of 4.345% compared to the one obtained in the previous step.
- The individual AP of several classes were improved such as apple black spot, apple healthy (leaves), apple mosaic virus, grapevine black spot, pear fire blight, and pear stony pit at 61.66%, 100%, 96.43%, 98.53%, 98.45%, and 96.66%.
- The last step before proposing any modification to the RFCN model is the utilization of different DL optimizers. SGD (with momentum) is used to train the model as the default DL optimizer. Subsequently, its performance is compared with that of Adam and RMSProp.
- After training the RFCN model using all three DL optimizers, it is found that SGD with momentum is the best DL optimization algorithm. Adam optimizer is unable to achieve a high mAP. Therefore, it did not effectively optimize the weights of the RFCN model. However, RMSProp has also achieved a lower mAP of 82.819%.
- Individual APs of several classes, including apple black rot, apple black spot, and pear canker are degraded by RMSProp.
- The best performance of the SGD optimizer demonstrates its generalizability in extracting the features of the healthy and disease classes and optimizing the weights of the RFCN. Therefore, it can be summarized that, for the NZDLPlantDisease-v1 dataset of healthy and diseased plant organs, the non-adaptive optimizer - SGD with momentum was quite successful compared to the adaptive optimization techniques RMSProp and Adam.
- To address one of the research gaps presented in the previous section, the RFCN model trained with a random normal initializer, using batch normalization and SGD with momentum optimizer is also successful in identifying multiple disease problems and detection of plant disease in different weather conditions (sunny, cloudy), as shown in Fig. 13 and Fig. 14, respectively.

Another important observation is that the pear scab class still attained a low AP of 5.06%. Although all steps until the application of various DL optimizers significantly improved the performance of RFCN in terms of a lower training and validation losses compared to the default configurations of the model. Still, the pear scab remained almost undetected or falsely identified. This result has provided a strong basis for the next steps of the research to focus on the architectural evaluation/modifications of the RFCN model.

E. PERFORMANCE ENHANCEMENT OF PEAR SCAB

There are two major goals for this step of the research. First, an improvement in the AP of pear scab, which is undetected after the application of several techniques explained earlier.

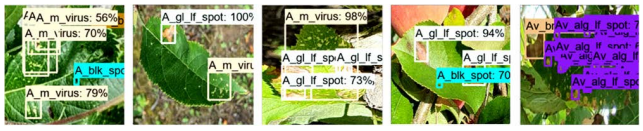


FIGURE 13. Examples of multiclass plant disease detection after the application of random normal initializer and batch normalization for the RFCN model.

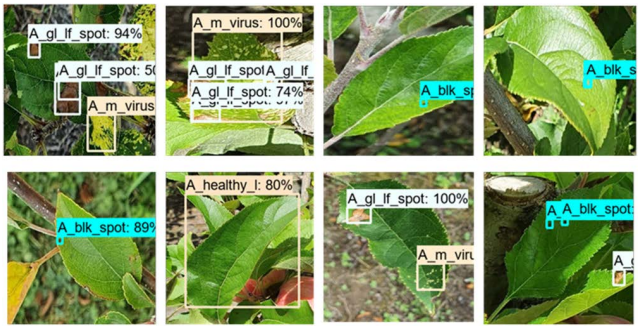


FIGURE 14. Examples of apple plant disease/healthy leaves in different environmental conditions.

Second, the high AP of the other 19 healthy/disease classes should be maintained. In this regard, the RFCN model has been investigated in two stages: position-sensitive score maps and enhanced/modified anchor box scale and aspect ratio.

One of the primary novelties of the RFCN model is the generation of position-sensitive score maps. The spatial bin configuration was set to 3×3 by default. It was empirically observed that the spots of pear scab were so small that the model could not extract its features and therefore could not be detected. This might be because none of the sub-regions of positive-sensitive score maps could match the pear scab for most of the testing images. Therefore, the position-sensitive region of interest (RoI) pool cannot vote for pear scab disease. In this regard, the first attempt is to increase the score maps using multiples of 3. The width and height of the 9×9 spatial grid have yielded satisfactory results and attained an mAP of 84.68%. Otherwise, with other spatial bins, such as 6×6, 12×12, and 15×15, a lower mAP of 82.819%, 82.041% and 82.59%, respectively are observed with the AP of the pear scab of 5.03%, 5.05%, and 5.28%. However, there is a slight difference in the total training and validation losses with 9×9 grids and the model has detected pear scab with an AP of 20.1%. Still, the DL model could detect the pear scab with a high AP. Furthermore, the RFCN trained with 9×9 spatial bins has disrupted the detection of apple black spots and achieved a lower AP of 49.69%. An example of the detection of apple black spot is presented in Fig. 15.

Another attempt has been made to solve this problem. The training images of the pear scab are magnified, and the RFCN model has been trained again. This is one of the ways to overlap the pear scab with 3×3 score maps. RFCN has successfully detected and localized both the apple black spot

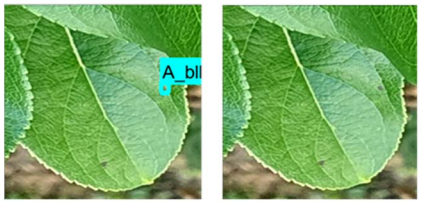


FIGURE 15. Detection outcomes for apple black spot using different positive sensitive score maps. (a) true positive results of apple black spot with 3 × 3 spatial bins, (b) false-negative detection with 9 × 9 spatial bins.

and the pear scab simultaneously. This indicates that the specific features of both disease classes are well extracted. The mAP is 87.394%, with the individual AP of the apple black spot and the pear scab at 59.85% and 94.43%, respectively. Few samples of the pear scab results are shown in Fig. 16. Moreover, a high AP of the rest of the classes is also maintained.

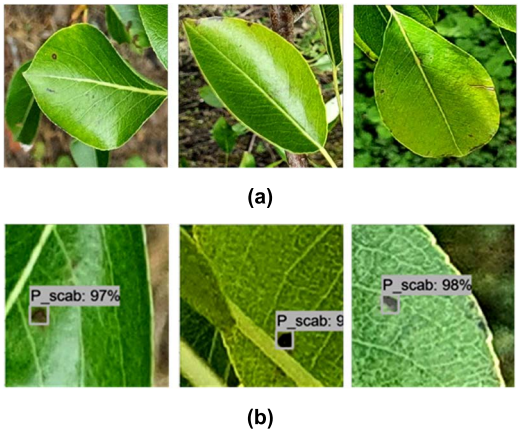


FIGURE 16. (a) Examples of false-negative results before zooming in the pear scab with 3 × 3 spatial bins (b) Examples of true positive results after zooming in the pear scab with 3 × 3 spatial bins

F. ENHANCEMENT OF ANCHOR BOXES

After a significant improvement in the average precision of the pear scab has been achieved in the previous step, the anchor boxes of the RFCN are enhanced. In this regard, the scale size and aspect ratios are modified to obtain an optimum anchor box that can provide an AP of more than 80% for each class. The summary of the results is as follows.

- Although the previous step has considerably improved the AP of the pear scab, a few classes also required attention towards further performance enhancement. For example, classes such as apple black rot, apple black spot, apple European canker, and pear healthy (leaves) have achieved an AP of less than 80%. During the annotation of the training images, it was empirically observed that the bounding-box coordinates of several classes varied. Therefore, different scales are tested to generate the anchor boxes.
- Hence, the addition of scales such as 64 × 64, 32 × 32, and 16 × 16 with the default scale sizes such as 128 x 128, 256 x 256, and 512 x 512, has significantly improved the mAP.
- On the other hand, an addition of a very small scale size such as 8 × 8 reduced the mAP. Similarly, a very large scale size of 1024 × 1024 could not contribute to attaining better mAP.
- Next, reciprocal aspect ratios are applied and the default aspect ratios of 1:2 and 2:1 are replaced. It is found that 1:4 and 4:1 have achieved an almost similar result with mAP of 87.33% with scale sizes of 16 × 16, 32 × 32, 64 × 64, 128 × 128, 256 × 256, and 512 × 512. Otherwise, none of the other combinations of reciprocal aspect ratios has shown noticeable results.
- Subsequently, the effects of the step-by-step/gradual enhancement of the aspect ratio are studied. In this regard, a small aspect ratio was started to add from 1:4 to default ratios of 1:2, 1:1, and 2:1, and enhanced scale sizes. After several experiments, it is found that the addition of the aspect ratio like 1:2, 1:1, 2:1, 3:1, and 4:1, has improved the training and testing performance of the model. The total training and validation loss from 0.4-0.515% and 0.4-0.8% have been reduced to almost 0.3-0.37% and 0.4-0.71%, respectively Furthermore, the individual loss of box classifier localization loss was reduced to almost 0.2% from 0.3%, as shown in Fig. 17. There is no sign of overfitting as the losses were converged, no abrupt rise in the validation loss was observed after the final iteration step, and there was a small difference between both training and validation losses.
- The feature extraction of the healthy and disease plant classes has been presented by t-distributed stochastic neighbor embedding (t-SNE) plots in Fig. 18. It can be seen for each of the healthy/defective classes trained by the final RFCN model that there is a high interclass distance separability, a small intraclass distance and grouped clusters have been created which were concentrated in their respective features. Furthermore, the effectiveness of the proposed modifications has been presented by comparing the t-SNE plot for the previous step and the default settings of the RFCN and Faster RCNN model after the application of the OT data augmentation method.
- It can be observed in Fig. 18 (b) that after the application of several techniques (presented in section III C-E), some of the features of the classes such as apple black rot, apple black spot, apple European canker, and pear healthy leaves were not well extracted and confused with the features of other apple and pear classes. Similarly, the t-SNE plots by the RFCN model after the OT data augmentation technique (Fig. 18 (c)) attained comparatively small interclass distances. There were several features of the apple black spot, pear fire blight, and pear healthy leaves, were not extracted and

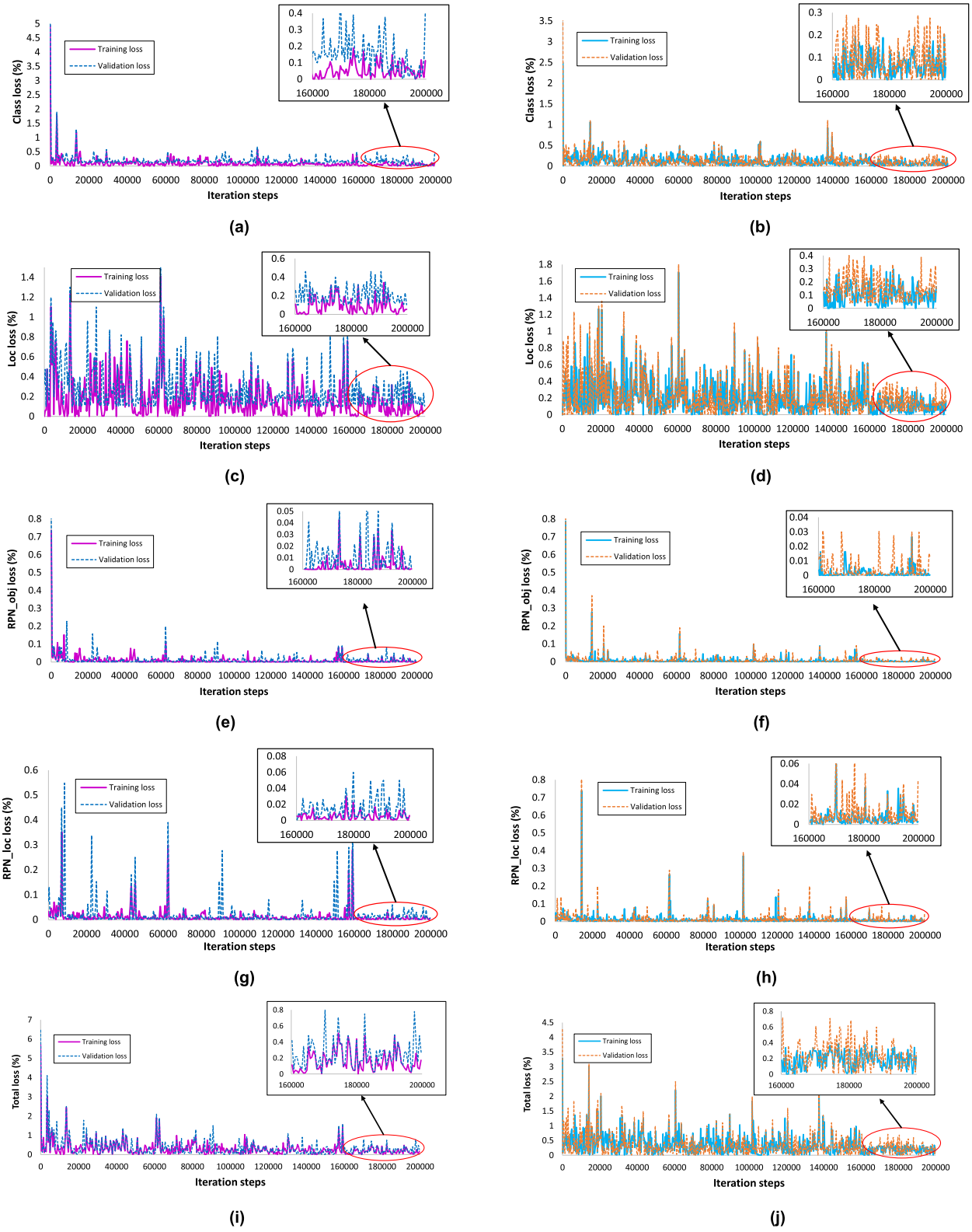


FIGURE 17. Training and validation losses for different anchor box configurations (a) box classifier classification loss with default scales and aspect ratio, (b) box classifier classification loss with modified anchor boxes, (c) box classifier localization loss with default scales and aspect ratio, (d) box classifier localization loss with modified anchor boxes, (e) RPN objectness loss with default scales and aspect ratio, (f) RPN objectness loss with modified anchor boxes, (g) RPN localization loss with default scales and aspect ratio, (h) RPN localization loss with modified anchor boxes, (i) total loss with default scales and aspect ratio, (j) total loss with modified anchor boxes.

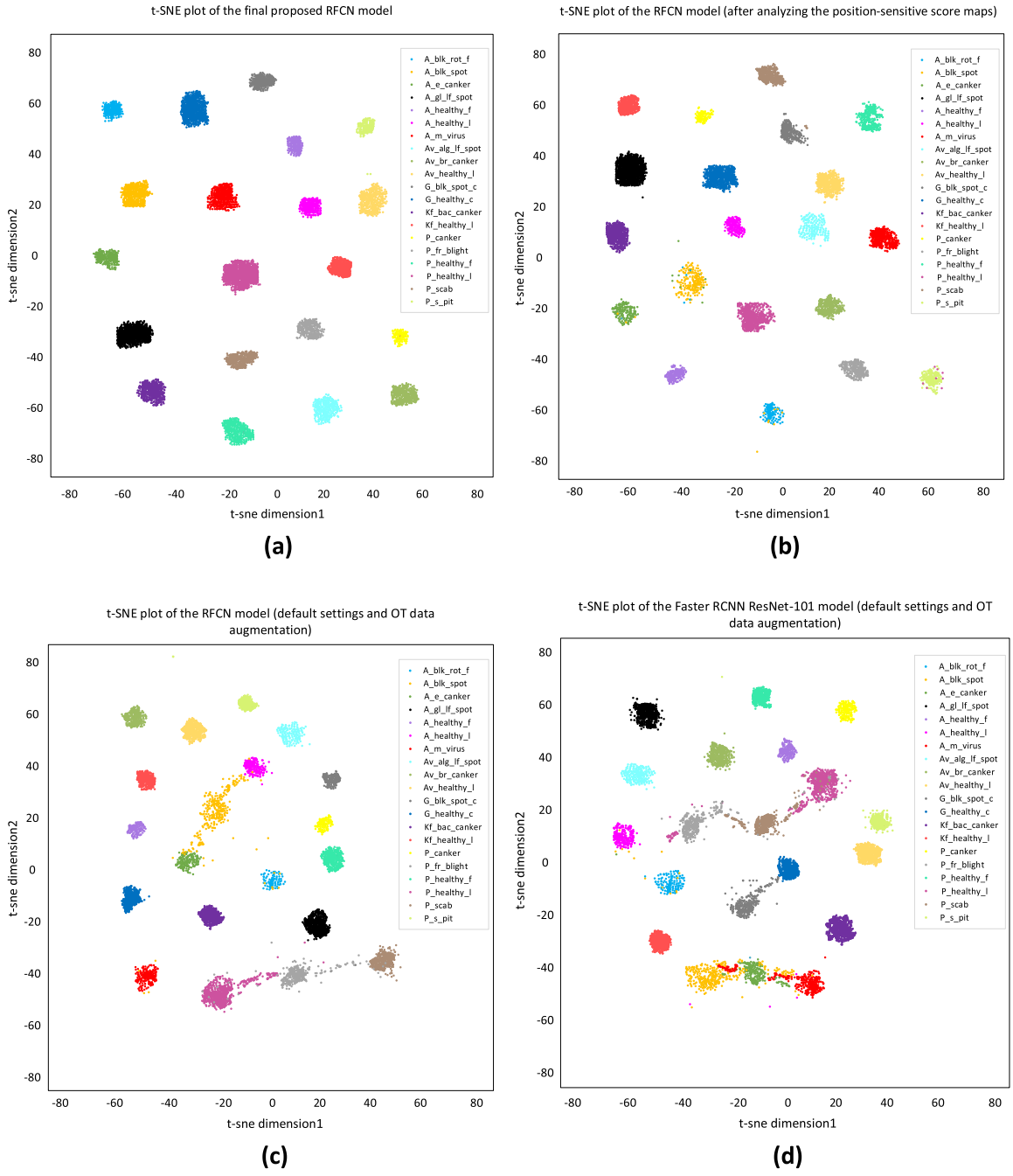


FIGURE 18. t-SNE plots of the proposed and existing methods. (a) final proposed RFCN model, (b) optimized model after the application of several weights optimization techniques and analyzing position-sensitive score maps, (c) RFCN model with default settings (after the application of translational augmentation techniques), (d) Faster RCNN ResNet-101 model with default settings (after the application of translational augmentation techniques).

confused with apple European canker, apple healthy leaves, pear fire blight, pear healthy leaves, and pear scab. Likewise, the Faster RCNN ResNet-101 model (Fig. 18 (d)) provided a degraded clustering performance as compared to the final RFCN model. The distinct features of apple black spot, apple European canker, apple mosaic virus, grapevine black spot, pear fire blight, pear healthy leaves, and pear scab were not made a proper cluster and confused with the features of other healthy/disease classes. This shows that the proposed modifications in the anchor boxes generate a significant difference in the feature extraction of the healthy and diseased plant classes.

- Moreover, the mAP is improved with a margin of 6.406% (Fig. 19). Also, a significant improvement in individual AP of classes such as apple black rot, apple

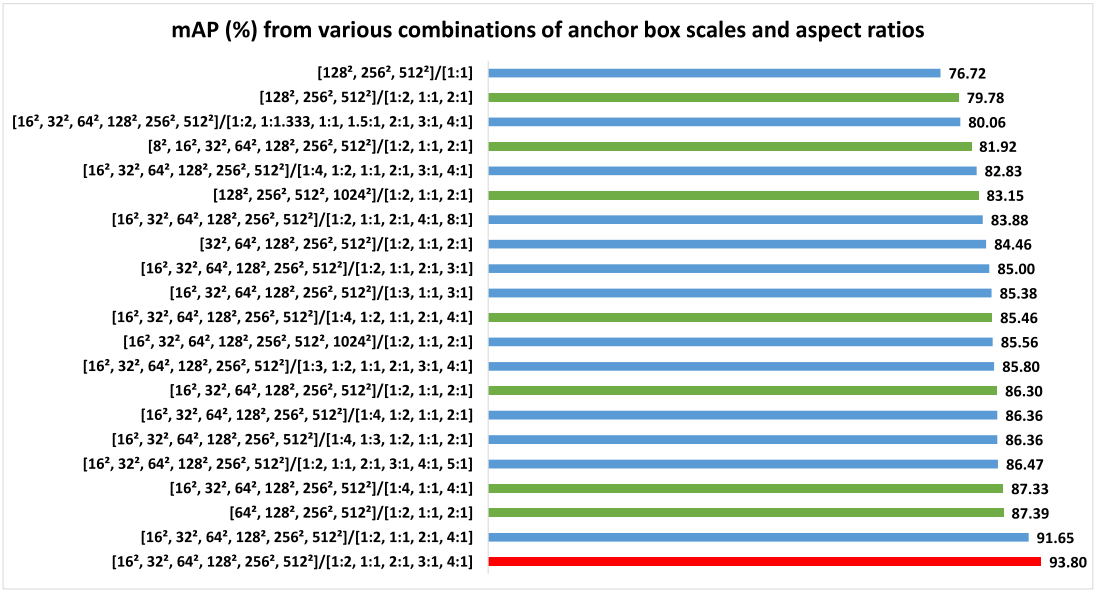


FIGURE 19. A summary of mAP with various specifications of anchor boxes.

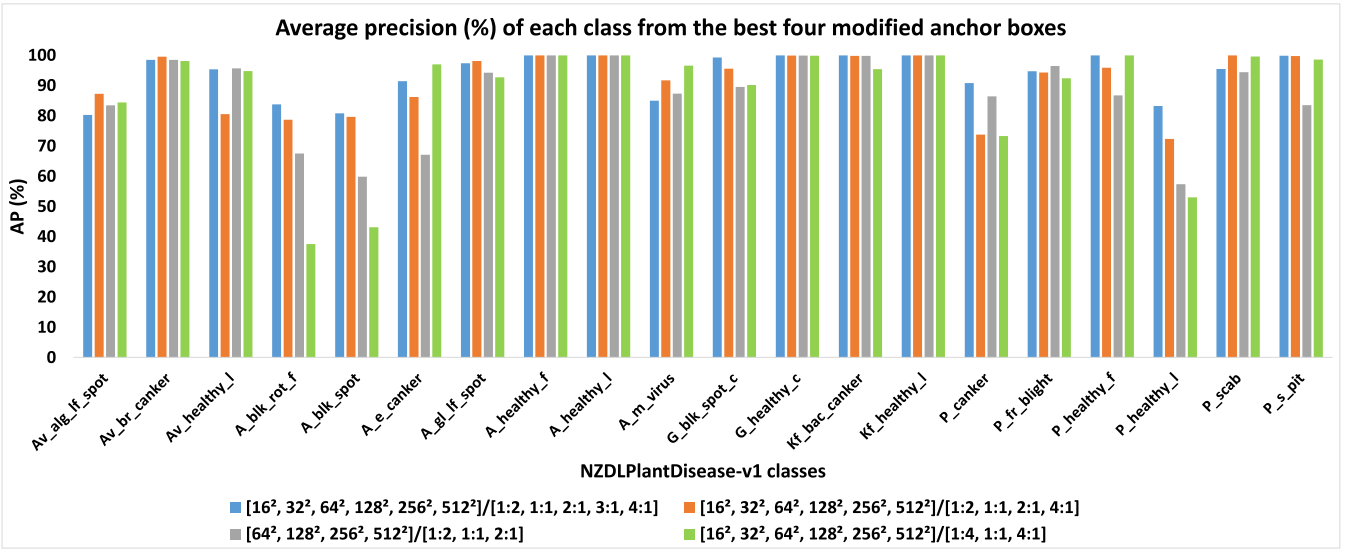


FIGURE 20. Average precision of each class by four prominent anchor box specifications.

black spot, apple European canker, pear healthy (leaves), and pear stony pit, as shown in Fig. 19.

- Other combinations of gradual addition of aspect ratios are examined, as shown in Fig. 19. One of the prominent combinations of 1:2, 1:1, 2:1, and 4:1 has attained a high mAP of 91.65
- In conclusion, the addition of various small scales like 16×16, 32×32, 64×64, and aspect ratios of 3:1 and 4:1 to the default anchor box have significantly improved the performance of the RFCN model with an mAP of 93.8. Furthermore, the AP of several classes is improved, and no class has achieved an AP of less than 80. Moreover, 12 classes have achieved a high AP of > 95, as shown in Fig. 20. The class-wise performance of the four prominent combinations of enhanced anchor boxes is presented in Fig. 20. A few examples of the false-negative results by the default anchor boxes, solved by the enhanced anchor boxes are presented in Fig. 21. A pictorial representation of the proposed modification of the anchor boxes is presented in Fig. 22.

G. OVERALL REMARKS ON THE PREVIOUS STEPS

A summary of the results presented from Section III-A to Section III-F is provided as under:

FIGURE 21. False-negative by default anchor box and true positive by final enhanced anchor box of classes including apple black rot, apple black spot, apple European canker, and pear healthy.

- The proposed deep learning (DL)-based method gradually enhanced the accuracy of plant disease detection from the step of comparison between deep learning architectures (Section III-A) to the enhancement of anchor boxes (Section III-F). The average precision of each class along with the mean average precision of the deep learning models are evaluated for each stage of the research. Each step has significance in terms of better training and testing results. The main reason for getting better results was that each step performed an in-depth analysis and identified a strong motivation for the subsequent steps to further improve mAP.
- For example, a comprehensive analysis of several DL models was performed to select the best-suited model. This selection was done and validated by using augmentation techniques. But the mAP obtained by all augmentation methods was significantly reduced. To cope with this problem, a category-wise comparison of the augmentation technique was performed, which gave us the best-suited technique for the selected application.
- Similarly, after the application of various techniques such as image resizers, interpolators, weight initializers, batch normalization, and deep learning optimizers, pear scab achieved unsatisfactory results. To attain a high AP of pear scab, the major novelty of the original RFCN model was analyzed, and the enhancement of the anchor boxes was attempted. In this way, the strong analyses of each step gave us the solid grounds for applying the following/next steps. A summary of the all steps including the best-selected method/model along with mAP is presented in Table 7.
- From Table 7, it can be concluded that all succeeding steps achieved a higher mAP, compared to its previous step. Furthermore, the most effective step in terms of improvement in the mAP was found to be the enhancement of the anchor boxes with an improvement of 6.406 compared to its earlier step.
- The effectiveness of the proposed approach has also been presented by the confusion matrix. For example, classes such as the apple black spot and pear

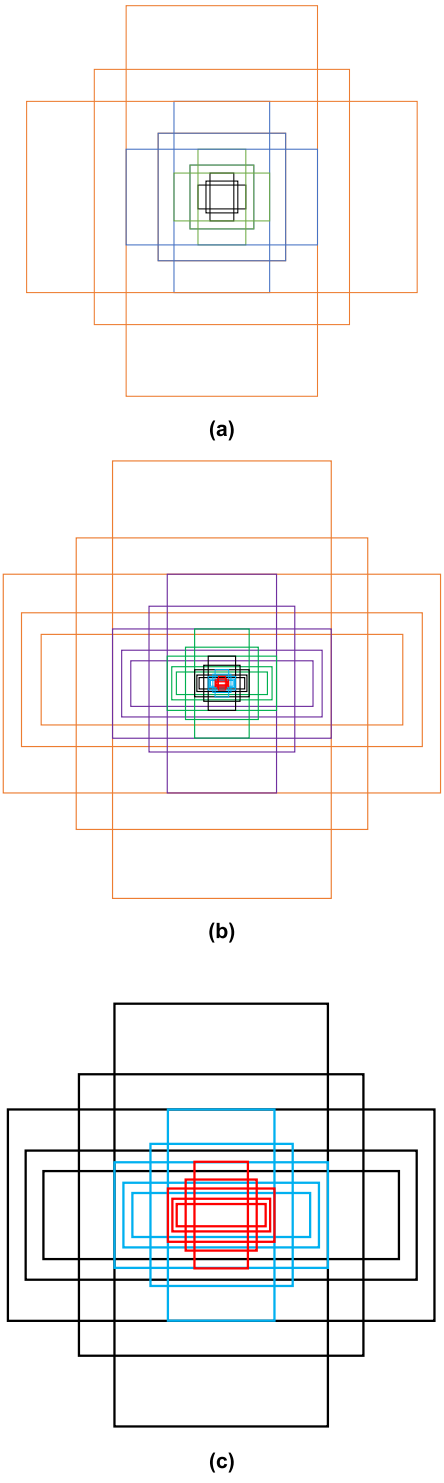


FIGURE 22. Anchor boxes for the RFCN model (a) present default anchor scales and aspect ratio, (b) present modified anchor boxes, (c) presents a zoomed version of 16 × 16 (red-colored boxes), 32 × 32 (blue colored boxes), and 64 × 64 (black colored boxes) scales sizes in the proposed version.

stony pit were confused with the healthy apple (leaves) and the fire blight, respectively, during the initial step of the methodology (Fig. 23 (a)). This can be verified by their detection results, already presented in

TABLE 7. Main findings of each step of the proposed methodology.

Steps	Best-obtained method/model	Mean average precision (%)
Step-a: Comparison between dl architectures	RFCN model	74.47
Step-b: Effects of data augmentation techniques	Original and translational changes in the images (OT)	76.76
Step-c: Effects of image resizers and interpolators	Fixed-shape resizer with bicubic interpolator	80.59
Step-d: Effects of weight initializers, batch normalization, and optimizers	Random normal (RN), batch normalization (BN), SGD with momentum (SGDm)	RN: 81.506, BN and SGDm: 85.94,
Step-e: Performance enhancement of pear scab	Position-sensitive score maps: 3x3, zoomed pear scab images	87.394
Step-f: Enhancement of anchor boxes	Scales: [16^2, 32^2, 64^2, 128^2, 256^2, 512^2] and aspect ratios: 1:2, 1:1, 2:1, 3:1, 4:1	93.80

Ground truth \ Predicted	A_blk_rot_f	A_blk_spot	A_e_canker	A_gl_lf_spot	A_healthy_f	A_healthy_l	A_m_virus	Av_alg_lf_spot	Av_br_canker	Av_healthy_l	G_blk_spot_c	G_healthy_c	Kf_bac_canker	Kf_healthy_l	P_canker	P_fr_blight	P_healthy_f	P_healthy_l	P_scab	P_s_pit
A_blk_rot_f	8	0	0	0	2	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0
A_blk_spot	0	13	0	5	0	7	2	0	0	1	0	0	0	0	0	0	0	2	2	0
A_e_canker	0	2	15	0	0	0	0	0	3	0	0	0	0	0	4	1	0	0	0	0
A_gl_lf_spot	0	5	0	46	1	0	0	2	0	10	0	0	0	0	0	0	0	0	1	0
A_healthy_f	0	0	0	0	20	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
A_healthy_l	0	0	0	0	0	16	0	0	0	1	0	0	0	0	0	0	0	1	0	0
A_m_virus	0	10	1	4	0	5	34	1	0	0	0	0	0	0	0	0	0	0	0	0
Av_alg_lf_spot	0	0	0	0	0	0	0	151	12	0	0	0	0	0	0	0	0	0	0	0
Av_br_canker	0	0	0	1	0	0	0	0	29	6	0	0	0	0	0	0	0	0	3	0
Av_healthy_l	0	0	0	0	0	1	0	0	0	17	0	0	0	0	0	0	0	0	0	0
G_blk_spot_c	0	0	0	0	0	0	0	0	0	0	12	1	0	0	0	0	0	0	0	0
G_healthy_c	0	0	0	0	0	0	0	0	0	0	1	15	0	0	0	1	0	0	0	0
Kf_bac_canker	0	1	0	0	0	0	0	0	0	0	0	0	21	0	0	0	0	0	0	0
Kf_healthy_l	0	0	0	0	0	0	0	0	0	0	0	0	0	27	0	0	0	0	0	0
P_canker	0	0	3	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0
P_fr_blight	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	25	0	0	0	0
P_healthy_f	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18	0	0	0
P_healthy_l	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	52	0	0
P_scab	0	6	0	0	0	0	0	0	1	2	0	0	0	0	0	1	0	3	18	0
P_s_pit	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	5	0	0	7

(a)

Ground truth \ Predicted	A_blk_rot_f	A_blk_spot	A_e_canker	A_gl_lf_spot	A_healthy_f	A_healthy_l	A_m_virus	Av_alg_lf_spot	Av_br_canker	Av_healthy_l	G_blk_spot_c	G_healthy_c	Kf_bac_canker	Kf_healthy_l	P_canker	P_fr_blight	P_healthy_f	P_healthy_l	P_scab	P_s_pit
A_blk_rot_f	47	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A_blk_spot	0	106	0	8	0	10	10	0	0	5	0	0	0	9	0	0	1	0	0	0
A_e_canker	0	0	93	5	1	1	0	0	2	0	0	2	0	0	1	0	0	0	0	0
A_gl_lf_spot	0	4	0	483	0	10	8	2	16	0	0	0	0	1	0	0	0	0	0	0
A_healthy_f	0	0	0	0	135	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0
A_healthy_l	0	0	0	0	0	178	0	0	0	0	0	0	0	1	0	2	0	0	0	0
A_m_virus	0	6	0	32	0	0	500	0	3	0	0	0	12	0	0	0	0	0	0	0
Av_alg_lf_spot	0	0	0	0	0	0	0	319	43	9	0	0	0	0	0	0	0	0	0	0
Av_br_canker	0	7	0	2	0	0	0	0	395	8	0	0	0	0	0	0	0	0	1	0
Av_healthy_l	0	0	0	0	0	10	0	0	0	145	0	0	0	0	0	0	1	1	0	0
G_blk_spot_c	0	0	0	0	0	0	0	0	0	0	86	11	0	0	0	0	0	0	0	0
G_healthy_c	0	0	0	0	0	0	0	0	2	0	0	130	0	0	0	0	0	0	0	0
Kf_bac_canker	0	0	0	0	0	0	0	0	0	0	0	0	109	0	0	0	0	0	0	0
Kf_healthy_l	0	0	0	0	0	0	0	0	0	0	0	0	0	213	0	0	0	0	0	0
P_canker	0	0	13	0	0	0	0	0	0	0	0	0	0	0	53	0	0	0	0	0
P_fr_blight	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	114	0	0	0	0
P_healthy_f	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	148	1	0	0
P_healthy_l	0	0	0	0	0	4	0	0	0	24	0	0	0	0	0	0	0	213	0	0
P_scab	0	4	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	3	95	0
P_s_pit	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	161

(b)

FIGURE 23. Confusion matrix for RFCN model (a) from the first step: training by original images with default settings, (b) from the last step: after the enhanced anchor boxes.

Figs. 9 (d) and (f). Consequently, a low recall of these classes was attained at 40.62 and 36.84, respectively. However, none of the classes suffered from a high number of wrong/missed classifications after the enhanced anchor boxes (Fig. 23 (b)), which led to significantly high mAP.

H. VALIDATION OF THE FINAL RESULTS

This study has also validated the results and the claims described in this article, in two ways. The first technique adopted is the stratified five-fold cross-validation, through which the dataset images of each class are folded five times so that the testing images in each fold are different from one-fold to another. This technique has been applied because of the class imbalance problem in the generated dataset and it avoids biased distribution in the dataset for each fold. The mAP obtained through all folds varied from 0.65% to 1.13% in the optimized configuration of the RFCN model, as compared to the final mAP. The first fold is considered as a default in which the modified/optimized RFCN model obtained 93.80%, while fold2, fold3, fold4, and fold5 attained mAP of 93.15%, 94.93%, 94.62%, and 93.09%, respectively.

To further evaluate the final mAP by the proposed method, the variance was calculated by the formula (31) and evaluated as 0.70157.

$$\text{Variance} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n - 1} \tag{31}$$

where n is the number of folds, x_i is the fold number (fold1, fold2, etc.), and $\bar{x}$ is the mean.

The next way to validate the claimed results is to test the optimized model on an external curated dataset. The original and translational/rotational change - OT augmentation method is also applied to that testing dataset. The mAP is 87.95%, which is only 5.85% lower than the final mAP obtained from the testing sub-dataset of the proposed dataset. Eleven classes, including avocado algal leaf spots, avocado branch cankers, avocado healthy, apple black rot, apple glomerella leaf spot, apple healthy (fruit), grapevine healthy, kiwifruit healthy, pear healthy (fruit), pear fire blight, and pear scab, are detected with a high AP of more than 90%. However, a few classes, such as pear canker, grapevine black spot, and pear stony pit, attained an AP of less than 80%. Therefore, the difference in mAP is obtained from that obtained by the testing images of the NZDLPlantDisease-v1 dataset. Examples of a few classes that achieved high and low AP from the external dataset are presented in Fig. 24.

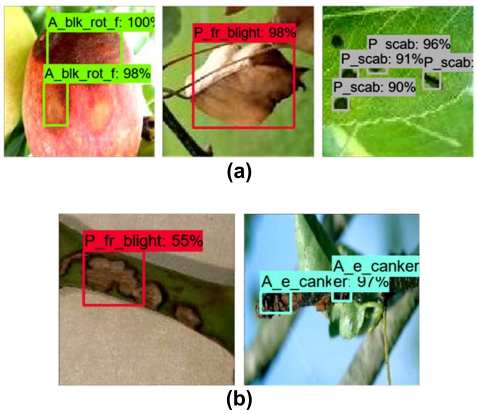


FIGURE 24. Results from an external dataset (a) True positive results of apple black rot, pear fire blight, and pear scab; (b) false positives of grapevine black spot and pear canker.

I. LIMITATIONS OF THE STUDY

Although the presented methodology has successfully detected the plant disease using the proposed dataset. Still, there are a few limitations of this study that can be taken into account in future research. The presence of disease in multiple organs of apple and pear has been considered for this research. Whereas, for grapevine, avocado, and kiwifruit, disease in only one plant organ has been considered. Moreover, only one disease class is presented for both grapevine and kiwifruit. Therefore, the proposed dataset should be further extended to get more insight into deep learning-based plant disease detection. Moreover, the validation of the modified/optimized model on an external generated dataset revealed that few of the classes did not achieve a high AP. One of the reasons could be the absence of diversity in the samples of those classes in the presented dataset. Furthermore, the dataset images from both sides of the plant organs should be considered. For example, the symptoms of the disease on the front and backside of the plant leaf could be included. This would generate more variety in the symptoms of plant disease. Also, all dataset images were collected from New Zealand horticultural fields. However, the dataset can be extended by capturing images of similar diseases in the same crops from horticultural fields in different countries. Moreover, the annotation was a bit tiring process due to the addition of the augmented images. Furthermore, as this article has addressed various practical problems, the most difficult among them was the detection of multiple diseases in a plant organ at a time. This task required even more time to correctly annotate the task. Therefore, it can be said that there is still a human intervention to use deep learning to perform complex task like plant disease detection.

IV. CONCLUSION AND FUTURE DIRECTIONS

This study addresses various research gaps in the identification of plant diseases based on deep learning. In this regard, a new dataset called NZDLPlantDisease-v1 is generated, and a DL-based approach is presented to detect and localize the disease in five of the most important New Zealand horticultural crops in terms of export value. After training and testing various DL architectures, the region-based fully convolutional network (RFCN) has achieved the highest mean average precision with and without the application of augmentation techniques. The proposed methodology consists of a comprehensive evaluation of various techniques impacted on the deep learning model that has not yet been explored for plant disease identification tasks. Furthermore, a modified/optimized version of the RFCN model is proposed by performing an in-depth analysis of position-sensitive score maps and anchor-box scales with aspect ratios. An improved mAP of 93.80% is achieved, which was 19.33% better than the default setting. The optimized RFCN includes training the model with a fixed-shape resizer with a bicubic interpolator, a random normal initializer, use of batch normalization, and SGD with a momentum optimizer. It is also observed that the translational/rotational augmentation method is the most suitable for obtaining satisfactory results. Furthermore, the addition of a 16×16, 32×32, 64×64 scales with an aspect ratio of 3:1 and 4:1 significantly improved the performance of the RFCN. The optimized/modified RFCN model has successfully answered research questions, including the detection of diseases in several plant organs, the presence of multiple diseases in one organ at a time, and the identification of diseases in different crops using the same trained DL model. Finally, the statements and results are validated by two different methods: stratified five-fold cross-validation and testing on an external dataset. These validation approaches demonstrate the significance and novelty of this study.

Furthermore, one of the advantages of this study is that different crops (selected for this research) have certain variations in their environments/backgrounds. Therefore, high average precision in each class shows an extended potential of deep learning technology for the detection of plant diseases, considering various challenges of the horticultural environment.

The idea/methodology proposed in this study can be utilized in several ways in future studies. A deep learning-based method can be embedded in automated/robotic systems to apply disease control techniques. For example, a fungicide spray can be applied to the defective parts of plants using a robotic manipulator. Furthermore, the diseases are treated differently based on the pathogen affecting the plants. Therefore, the research question related to the detection of multiple classes of plant diseases (suffering from different diseases at a time) [50] could be useful for implementing a cost-effective protection system. For instance, black spots on apples are normally treated with a fungicide spray, whereas no such treatment is available for apple viruses [51]. Hence, this research will be helpful for growers to take appropriate treatment measures after detecting multiple plant diseases in an organ.

In addition, various tasks can be performed to further enhance research on DL-based plant diseases. For instance, advanced data augmentation techniques, including super-resolution convolutional neural networks (SRCNNs) and super-resolution generative adversarial networks (SRGANs), can be explored. Moreover, segmentation-based DL models can be leveraged and modified by using the generated dataset. Furthermore, the performance metrics presented in [52] can also be explored to perform a more in-depth analysis of the multi-label plant disease detection problem. Some other research ideas can be explored to further strengthen the research on DL-based solutions for agricultural problems. For example, the sensitivity analysis (like the one performed for a teleoperation system to examine the effects of important parameters on the system performance [53]) can be performed for the DL models to implement various agricultural operations. Moreover, the layer-wise output of the well-known DL models could be visualized to modify the hidden layer. The comparison of CNNs with CapsuleNet models can also be emphasized, as this class of machine learning is being explored for various object detection problems [54].

AVAILABILITY OF DATA

The dataset generated and analyzed during the current study is available in the GitHub repository https://github.com/kmarif/NZDLPlantDisease-v1.

ACKNOWLEDGMENT

The authors are thankful to Dr. Carl Mesarich (School of Agriculture and Environment, Massey University) for providing the common and scientific names of plant diseases. They are also grateful to Fakhia Hammad and Muhammad Taha for their help with the annotation of the dataset images.

REFERENCES

[1] A. G. Aitken and I. J. Warrington. (2020). *Fresh Facts*. Plant & Food Research. [Online]. Available: https://www.freshfacts.co.nz/files/freshfacts-2020.pdf

[2] J. Liu and X. Wang, "Plant diseases and pests detection based on deep learning: A review," *Plant Methods*, vol. 17, no. 1, pp. 1–18, Dec. 2021.

[3] M. H. Saleem, J. Potgieter, and K. M. Arif, "Automation in agriculture by machine and deep learning techniques: A review of recent developments," *Precis. Agricult.*, vol. 22, no. 6, pp. 2053–2091, 2021.

[4] V. S. Dhaka, S. V. Meena, G. Rani, D. Sinwar, K. Kavita, M. F. Ijaz, and M. Woźniak, "A survey of deep convolutional neural networks applied for prediction of plant leaf diseases," *Sensors*, vol. 21, no. 14, p. 4749, Jul. 2021.

[5] M. Alessandrini, R. C. F. Rivera, L. Falaschetti, D. Pau, V. Tomaselli, and C. Turchetti, "A grapevine leaves dataset for early detection and classification of esca disease in vineyards through machine learning," *Data Brief*, vol. 35, Apr. 2021, Art. no. 106809.

[6] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers Plant Sci.*, vol. 7, p. 1419, Sep. 2016.

[7] M. Brahimi, K. Boukhalfa, and A. Moussaoui, "Deep learning for tomato diseases: Classification and symptoms visualization," *Appl. Artif. Intell.*, vol. 31, no. 4, pp. 299–315, 2017.

[8] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, "A comparative study of fine-tuning deep learning models for plant disease identification," *Comput. Electron. Agric.*, vol. 161, pp. 272–279, Jun. 2019.

[9] O. O. Abayomi-Alli, R. Damaševičius, S. Misra, and R. Maskeliūnas, "Cassava disease recognition from low-quality images using enhanced data augmentation model and deep learning," *Exp. Syst.*, vol. 38, no. 7, Nov. 2021, Art. no. e12746.

[10] L. Bi and G. Hu, "Improving image-based plant disease classification with generative adversarial network under limited training set," *Frontiers Plant Sci.*, vol. 11, p. 1945, Dec. 2020.

[11] D. P. Hughes and M. Salathe, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," 2015, *arXiv:1511.08060*.

[12] A. Abbas, S. Jain, M. Gour, and S. Vankudothu, "Tomato plant disease detection using transfer learning with C-GAN synthetic images," *Comput. Electron. Agricult.*, vol. 187, Aug. 2021, Art. no. 106279.

[13] W. Albattah, M. Nawaz, A. Javed, M. Masood, and S. Albahli, "A novel deep learning method for detection and classification of plant diseases," *Complex Intell. Syst.*, vol. 8, no. 1, pp. 507–524, Feb. 2022.

[14] K. Kc, Z. Yin, M. Wu, and Z. Wu, "Depthwise separable convolution architectures for plant disease classification," *Comput. Electron. Agricult.*, vol. 165, Oct. 2019, Art. no. 104948.

[15] M. H. Saleem, J. Potgieter, and K. M. Arif, "Plant disease classification: A comparative evaluation of convolutional neural networks and deep learning optimizers," *Plants*, vol. 9, no. 10, p. 1319, Oct. 2020.

[16] M. H. Saleem, S. Khanchi, J. Potgieter, and K. M. Arif, "Image-based plant disease identification by deep learning meta-architectures," *Plants*, vol. 9, no. 11, p. 1451, Oct. 2020.

[17] F. Alvaro, Y. Sook, K. Sang, and P. Dong, "A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition," *Sensors*, vol. 17, no. 9, p. 2022, 2017.

[18] A. Karlekar and A. Seal, "SoyNet: Soybean leaf diseases classification," *Comput. Electron. Agricult.*, vol. 172, May 2020, Art. no. 105342.

[19] C. K. Sunil, C. D. Jaidhar, and N. Patil, "Cardamom plant disease detection approach using EfficientNetV2," *IEEE Access*, vol. 10, pp. 789–804, 2022.

[20] U. P. Singh, S. S. Chouhan, S. Jain, and S. Jain, "Multilayer convolution neural network for the classification of mango leaves infected by anthracnose disease," *IEEE Access*, vol. 7, pp. 43721–43729, 2019.

[21] N. Yao, F. Ni, Z. Wang, J. Luo, W.-K. Sung, C. Luo, and G. Li, "L2MXception: An improved xception network for classification of peach diseases," *Plant Methods*, vol. 17, no. 1, pp. 1–13, Dec. 2021.

[22] J. Liu and X. Wang, "Tomato diseases and pests detection based on improved Yolo V3 convolutional neural network," *Frontiers Plant Sci.*, vol. 11, p. 898, Jun. 2020.

[23] A. Fuentes, S. Yoon, M. H. Lee, and D. S. Park, "Improving accuracy of tomato plant disease diagnosis based on deep learning with explicit control of hidden classes," *Frontiers Plant Sci.*, vol. 12, p. 2938, Dec. 2021.

[24] J. Sun, Y. Yang, X. He, and X. Wu, "Northern maize leaf blight detection under complex field environment based on deep learning," *IEEE Access*, vol. 8, pp. 33679–33688, 2020.

[25] P. Wang, T. Niu, Y. Mao, Z. Zhang, B. Liu, and D. He, "Identification of apple leaf diseases by improved deep convolutional neural networks with an attention mechanism," *Frontiers Plant Sci.*, vol. 12, p. 1997, Sep. 2021.
[26] X. Xie, Y. Ma, B. Liu, J. He, S. Li, and H. Wang, "A deep-learning-based real-time detector for grape leaf diseases using improved convolutional neural networks," *Frontiers Plant Sci.*, vol. 11, p. 751, Jun. 2020.
[27] J. G. Barbedo, "Deep learning applied to plant pathology: The problem of data representativeness," *Tropical Plant Pathol.*, vol. 47, pp. 85–94, Aug. 2021.
[28] E. Mwebaze, T. Gebru, A. Frome, S. Nsumba, and J. Tusubira, "ICassava 2019 fine-grained visual categorization challenge," 2019, *arXiv:1908.02900*.
[29] P. K. Sethy, N. K. Barpanda, A. K. Rath, and S. K. Behera, "Deep feature based Rice leaf disease identification using support vector machine," *Comput. Electron. Agricult.*, vol. 175, Aug. 2020, Art. no. 105527.
[30] H. T. Rauf, B. A. Saleem, M. I. U. Lali, M. A. Khan, M. Sharif, and S. A. C. Bukhari, "A citrus fruits and leaves dataset for detection and classification of citrus diseases through machine learning," *Data Brief*, vol. 26, Oct. 2019, Art. no. 104340.
[31] T. Wiesner-Hanks, E. L. Stewart, N. Kaczmar, C. DeChant, H. Wu, R. J. Nelson, H. Lipson, and M. A. Gore, "Image set for deep learning: Field images of maize annotated with disease symptoms," *BMC Res. Notes*, vol. 11, no. 1, pp. 1–3, Dec. 2018.
[32] J. Jepkoech, D. M. Mugo, B. K. Kenduiywo, and E. C. Too, "Arabica coffee leaf images dataset for coffee leaf disease detection and classification," *Data Brief*, vol. 36, Jun. 2021, Art. no. 107142.
[33] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 21–37.
[34] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 91–99.
[35] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 379–387.
[36] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.
[37] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10781–10790.
[38] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
[39] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.
[40] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.
[41] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.
[42] S. Ruder, "An overview of gradient descent optimization algorithms," 2016, *arXiv:1609.04747*.
[43] G. Hinton, N. Srivastava, and K. Swersky, "Neural networks for machine learning," *Coursera, Video Lectures*, vol. 264, no. 1, pp. 2146–2153, 2012.
[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[45] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, no. 2, pp. 1–25, 2012.
[46] H. He and Y. Ma, *Imbalanced Learning: Foundations, Algorithms, and Applications*. Hoboken, NJ, USA: Wiley, 2013.
[47] J. G. A. Barbedo, "Factors influencing the use of deep learning for plant disease recognition," *Biosyst. Eng.*, vol. 172, pp. 84–91, Aug. 2018.
[48] O. Rukundo, "Effects of image size on deep learning," 2021, *arXiv:2101.11508*.
[49] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, "How does batch normalization help optimization?" in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–11.
[50] M. B. Riley, M. R. Williamson, and O. Maloy, "Plant disease diagnosis," *Plant Health Instructor*, vol. 10, 2002, doi: 10.1094/PHI-I-2002-1021-01.
[51] S. U. Nabi, G. Madhu, G. P. Rao, and V. K. Baranwal, "Development of multiplex RT-PCR assay for simultaneous detection of four viruses infecting apple (Malus domestica)," *Lett. Appl. Microbiol.*, vol. 74, no. 4, pp. 586–592, 2022.
[52] X.-Z. Wu and Z.-H. Zhou, "A unified view of multi-label performance measures," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3780–3788.
[53] R. Uddin, M. H. Saleem, and J. Ryu, "Parametric sensitivity analyses for perceived impedance in haptic teleoperation," *Int. J. Control, Autom. Syst.*, vol. 17, no. 8, pp. 2083–2096, Aug. 2019.
[54] M. L. Mekhalfi, M. B. Bejiga, D. Soresina, F. Melgani, and B. Demir, "Capsule networks for object detection in UAV imagery," *Remote Sens.*, vol. 11, no. 14, p. 1694, Jul. 2019.

MUHAMMAD HAMMAD SALEEM received the B.E. and M.E. degrees in electrical engineering from the NED University of Engineering and Technology, Karachi, Pakistan, in 2016 and 2018, respectively. He is currently pursuing the Ph.D. degree with the School of Food and Advanced Technology (SF&AT), Massey University, Auckland, New Zealand. Before starting his Ph.D., he was a Lecturer with the Department of Electrical Engineering, NED University of Engineering and Technology. His research interests include artificial intelligence, deep learning, computer vision, machine learning, agricultural robotics, haptics, and teleoperation.

JOHAN POTGIETER is currently a Professor of robotics at the Massey AgriFood (MAF) Digital Laboratory. He has a well-established reputation for his work in educational robotics and has been inducted into the World Robotics Education and Competition (REC) Foundation Hall of Fame. His current research interests include additive manufacturing and advanced mechatronics/robotics with applications in medicine and automotive related areas.

KHALID MAHMOOD ARIF (Senior Member, IEEE) received the B.Sc. degree in mechanical engineering from the University of Engineering and Technology, Lahore, Pakistan, in 2000, the M.Eng. degree in engineering synthesis from The University of Tokyo, Japan, in 2004, and the Ph.D. degree in mechanical engineering from Purdue University, West Lafayette, IN, USA, in 2011. From 2007 to 2011, he was a Fulbright Scholar at Purdue University. He is currently a Senior Lecturer in mechatronics and robotics at Massey University, Auckland. Prior to joining Massey University, in 2012, he was an Assistant Professor with the Department of Mechatronics and Control Engineering, University of Engineering and Technology, Lahore. He is also the Director of the Sensors and Smart Systems Laboratory. His research interests include sensors, the IoT, robotics, and additive manufacturing.

Chapter 8 A Weight Optimization-based Transfer Learning Approach for Plant Disease Detection of New Zealand Vegetables

This chapter contains content from the following article.

M. H. Saleem, J. Potgieter, and K. M. Arif, "A weight optimization-based transfer learning approach for plant disease detection of New Zealand vegetables," *Frontiers in Plant Science*, vol. 13, 2022. https://doi.org/10.3389/fpls.2022.1008079

frontiers | Frontiers in Plant Science

TYPE Original Research
PUBLISHED 25 October 2022
DOI 10.3389/fpls.2022.1008079

OPEN ACCESS

EDITED BY
Yiannis Ampatzidis,
University of Florida, United States

REVIEWED BY
Hamidreza Bolhasani,
Islamic Azad University, Iran
Chengcheng Chen,
Jilin University, China

*CORRESPONDENCE
Khalid Mahmood Arif
k.arif@massey.ac.nz

SPECIALTY SECTION
This article was submitted to Technical Advances in Plant Science, a section of the journal Frontiers in Plant Science

RECEIVED 31 July 2022
ACCEPTED 22 September 2022
PUBLISHED 25 October 2022

CITATION
Saleem MH, Potgieter J and Arif KM (2022) A weight optimization-based transfer learning approach for plant disease detection of New Zealand vegetables. *Front. Plant Sci.* 13:1008079. doi: 10.3389/fpls.2022.1008079

A weight optimization-based transfer learning approach for plant disease detection of New Zealand vegetables

Muhammad Hammad Saleem[1], Johan Potgieter[2] and Khalid Mahmood Arif[1]*

[1]Department of Mechanical and Electrical Engineering, School of Food and Advanced Technology, Massey University, Auckland, New Zealand, [2]Massey AgriFood Digital Lab, Massey University, Palmerston North, New Zealand

Deep learning (DL) is an effective approach to identifying plant diseases. Among several DL-based techniques, transfer learning (TL) produces significant results in terms of improved accuracy. However, the usefulness of TL has not yet been explored using weights optimized from agricultural datasets. Furthermore, the detection of plant diseases in different organs of various vegetables has not yet been performed using a trained/optimized DL model. Moreover, the presence/detection of multiple diseases in vegetable organs has not yet been investigated. To address these research gaps, a new dataset named NZDLPlantDisease-v2 has been collected for New Zealand vegetables. The dataset includes 28 healthy and defective organs of beans, broccoli, cabbage, cauliflower, kumara, peas, potato, and tomato. This paper presents a transfer learning method that optimizes weights obtained through agricultural datasets for better outcomes in plant disease identification. First, several DL architectures are compared to obtain the best-suited model, and then, data augmentation techniques are applied. The Faster Region-based Convolutional Neural Network (RCNN) Inception ResNet-v2 attained the highest mean average precision (mAP) compared to the other DL models including different versions of Faster RCNN, Single-Shot Multibox Detector (SSD), Region-based Fully Convolutional Networks (RFCN), RetinaNet, and EfficientDet. Next, weight optimization is performed on datasets including PlantVillage, NZDLPlantDisease-v1, and DeepWeeds using image resizers, interpolators, initializers, batch normalization, and DL optimizers. Updated/optimized weights are then used to retrain the Faster RCNN Inception ResNet-v2 model on the proposed dataset. Finally, the results are compared with the model trained/optimized using a large dataset, such as Common Objects in Context (COCO). The final mAP improves by 9.25% and is found to be 91.33%. Moreover, the robustness of the methodology is demonstrated by testing the final model on an external dataset and using the stratified k-fold cross-validation method.

KEYWORDS

convolutional neural networks, deep learning, transfer learning, optimization algorithms, cross-validation, plant disease detection

Introduction

According to the guidelines of the World Health Organization (WHO) and the Food and Agriculture Organization (FAO), fruit and vegetable consumption should be greater than 400 g/day to reduce the risk of heart disease, high blood pressure, and stroke (Who and Consultation, 2003). Therefore, adequate food supply must be achieved for human well-being. In this regard, food security is essential to ensure food demand. Plant diseases are a substantial threat to food security (Gui et al., 2021) and affect crop productivity (Liu and Wang, 2021). They also influence the quality of agricultural products and are a source of economic loss.

Horticultural products are among the main contributors to New Zealand (NZ) economy. According to a recent report by Plant and Food Research, the export value of fresh and processed vegetables was estimated to be NZ$724.5m in 2020 (Aitken and Warrington, 2020). Among the most prominent vegetables, onions, peas, potatoes, squash, sweetcorn, and beans were exported in amounts of NZ$147.6m, NZ$115.4m, NZ$106.9m, NZ$79.2m, NZ$47.6m, and NZ$42.0m, respectively. The highest expenditures for imported vegetables were observed for preserved tomatoes and frozen potatoes at around NZ$35.8m and NZ$34.9m, respectively. A total of NZ$1.29b was recorded for the vegetables consumed for domestic use, such as potatoes, tomatoes, and brassicas (broccoli, cabbage, and cauliflower). Furthermore, the largest horticultural land (among the vegetables) was used for potatoes around 10,417 ha, 6,530 ha for squash, 5,296 ha for onion, and 4,890 ha for peas and beans (Aitken and Warrington, 2020). These statistics highlight the importance of NZ vegetables to the country's income. Therefore, the challenges faced by the horticultural industry should be addressed to increase the export value of horticultural products further.

Plant disease detection is an important task in the application of control treatments to the affected plants. However, the timely and precise identification of plant diseases is challenging. This is due to the similarity in the occurrence of the disease in different plant species. Therefore, the traditional methods for recognizing plant diseases have been replaced by methods based on machine learning (ML). Among these techniques, deep learning (DL) has gained considerable attention from the scientific community in recent years, and numerous DL methods, architectures, and approaches have been proposed. However, current literature leaves a significant margin for further investigating the strength of DL-based plant disease identification in various ways.

The research done so far for plant disease detection has covered various aspects of deep learning, such as DL-based visualization techniques to identify the spots of plant disease (Brahimi et al., 2018; Brahimi et al., 2019) and the application of the latest data augmentation techniques (Bi and Hu, 2020; Abbas et al., 2021). Furthermore, some studies have focused on modifications in the hidden layers of neural networks (Kamal et al., 2019; Liang et al., 2019; Liu and Wang, 2020), improvement in the feature fusion module (Bao et al., 2022), and the addition of attention modules (Wang et al., 2021a; Bao et al., 2022). Moreover, few articles have addressed the real-time assessment of the DL model for the recognition of plant disease (Xie et al., 2020; Wang et al., 2021b) and the evaluation of the usefulness of deep learning optimizers (Saleem et al., 2020).

Transfer learning (TL) is a successful technique for improving the performance of DL models. In TL, the knowledge gained from the pre-trained weights of a large dataset is utilized to extract the specific features of the dataset that contains new classes. A few studies have presented the significance of TL in plant disease identification. For example, the effectiveness of the fine-tuning technique was shown for plant disease classification (Brahimi et al., 2017; Too et al., 2019; Hassan et al., 2021), using pre-trained weights in a large dataset called ImageNet. Another study evaluated TL on synthetic images (generated by a generative adversarial network) and real images (Abbas et al., 2021). Although these articles presented the importance of fine-tuning, the selected dataset had a plain background/controlled environment. Only one study has performed a plant disease identification task in a real agricultural environment using a transfer learning approach (Chen et al., 2020a). Another study considered field conditions for the detection of diseases in maize plants using three models inspired by Inception-v3 (Haque et al., 2022). The performance of the proposed model was better than that of the pre-trained DL models. However, the DL architecture required a longer computation time. A study presented eggplant disease detection using pre-trained weights of the visual geometry group (VGG) model and created a new dataset with a combination of controlled/laboratory and uncontrolled/real agricultural environments (Krishnaswamy Rangarajan and Purushothaman, 2020). Another study presented TL by combining the DenseNet model and Inception module for rice plant diseases under actual conditions (Chen et al., 2020c). A two-phase approach to applying TL was presented by (Chen et al., 2020b). The first phase was dedicated to training from scratch for the new layers and using pre-trained weights for the rest of the layers, whereas the second phase consisted of retraining the model on the selected dataset using weights obtained in the previous stage. Although a comprehensive analysis was provided, the comparison of the proposed method with other modified versions of the DL architecture could further strengthen the quality of the work. In a similar style, a recent study proposed a transfer-learning-based deep feature descriptor (Fan et al., 2022). The authors demonstrated the effectiveness of the approach on different datasets and showed the novelty of this work. Another study presented the significance of TL in tomato plant disease identification (Thangaraj et al., 2021). In this work, a few of the models

were included for comparative analysis with the proposed model. A study evaluated the TL by splitting the data into the source and target domains (Argüeso et al., 2020). This study provided a new approach to exploring TL-based techniques. However, it should be validated in other datasets related to plant diseases. Another study evaluated TL based on domain splitting and analyzed semi-supervised iterations and few-shot parameters (Rukundo, 2021). This research presented validation using three domain splits of the same dataset, whereas a more comprehensive analysis could be included, such as considering different datasets with diverse environments. An article presented a multitask TL approach to identify rice and wheat plant diseases at the same time, compared to other techniques including a single task, a reuse model, and some DL models (Jiang et al., 2021). Comparison of TL using pre-trained weights on a large dataset consisting of general objects and a dataset related to plant recognition was presented by (Lee et al., 2020). This proved to be an innovative method to demonstrate the importance of transfer learning. It was also observed that a more in-depth analysis could be performed using training profiles.

The literature references presented above have considerable scope for exploring transfer learning methods to improve plant disease detection. The use of image resizers, weight initialization, and DL optimization methods have not been explored for transfer learning purposes. In addition, weight optimization has also not been performed using the datasets related to the same and different agricultural operations. Moreover, most articles have addressed the recognition of plant diseases in a single vegetable under real agricultural conditions. However, the effectiveness of deep transfer learning has not yet been evaluated for different vegetables using the same optimized DL model.

This article addresses several concerns regarding deep transfer-learning-based plant disease detection. The first research question is how well can deep learning detect plant diseases in various organs of vegetables? Can transfer learning be applied for the simultaneous identification of multiple diseases in various vegetable organs at the same time? How do the different methods related to image processing and weight optimization affect the performance of deep transfer learning? Can the weights obtained from the agricultural datasets contribute to improving the performance of the DL model? Connected to the previous question, what are the circumstances to obtain improvement in terms of the dataset conditions? Finally, can the highlighted questions be validated by testing on an external dataset with the same classes and retaining a high mean average precision?

To answer these questions, this article presents a new dataset (the second version in a series of datasets containing plant diseases in various NZ horticultural crops), named NZDLPlantDisease-v2. The generated dataset contains diseases in eight important NZ vegetables, including potatoes, tomatoes, beans, peas, kumara, and brassicas (broccoli, cabbage, and cauliflower). The dataset also contains diseases in different plant organs (leaves, stems, and vegetables), along with multiple diseases in plant organs at a time. Moreover, this study proposes transfer-learning-based plant disease identification. For this purpose, the weights of the best-obtained DL architecture are optimized using a framework consisting of various techniques. These methods include the evaluation of image resizers, interpolators, weight initializers, batch normalization, and DL optimizers. This framework has been derived from our recently presented work on weed detection (Saleem et al., 2022b). Furthermore, optimized weights are obtained using agricultural (similar and/or other applications) datasets. In this regard, three datasets are selected, two of which are related to plant diseases in both laboratory and real field environments, and the last is a weed-related dataset. In addition, weight optimization is performed on a large dataset called common objects in context (COCO). The mean average precisions of the weights obtained using the agricultural and large datasets are compared. Finally, the usefulness of the research is demonstrated by testing on an external dataset and using the stratified k-fold cross-validation method. Hence, this article provides new insights into DL-based plant disease recognition rather than overly explored open-source datasets such as PlantVillage.

The main contributions of this research are as follows: (1) a new dataset is presented consisting of eight prominent vegetables of the NZ and it has been made publicly available to the scientific community; (2) several conditions and problems of the real horticultural field have also been considered, including the presence of disease in various organs of the vegetable plants, different environmental conditions, and the occurrence of multiple diseases in a vegetable plant organ at a time; (3) a new way of exploiting various techniques has been provided by the integration of weight optimization and transfer learning methods; (4) a high mean average precision (mAP) has been achieved by comparing with the results obtained through pre-trained weights of the COCO dataset; an average precision (AP) of >80% for each class has also been achieved; (5) the robustness of the presented work has been validated by two methods: testing on an externally generated dataset in different agricultural conditions and using the stratified five-fold cross-validation method.

Materials and methods

A transfer learning-based approach

The proposed approach is based on transfer learning for the extraction of features of plant diseases. The weights of the best-obtained DL model are updated using the optimized weights from the agricultural dataset. Performance optimization techniques are applied as a bridge to update/optimize the

weights of the DL model. The proposed methodology is divided into two phases. Each phase comprises two steps, and the final validation/effectiveness of the approach is presented using two different methods.

The first step in the initial phase of the research was the generation of the dataset called NZDLPlantDisease-v2. The proposed dataset contains eight important vegetables from New Zealand, including 28 healthy and diseased classes. This dataset contains various practical problems in the horticultural field. The second step was a comparative evaluation of various DL meta-architectures to obtain the best DL model in terms of training, validation losses, and mean average precision (mAP). This step of the proposed methodology uses the TL technique by training the models using pre-trained weights on the COCO dataset to extract the basic features. The performance of the best-obtained model was enhanced by applying category-wise data augmentation techniques, including color change (brightness, contrast, and sharpness), translational and rotational changes, and the addition of noise to the color change.

The second phase began with the selection of agricultural datasets. In this regard, two datasets of healthy and defective plant organs and a dataset of weed images were selected. One of the selected plant disease datasets has a plain background, whereas the other has a natural/complex agricultural background. This was done to analyze the effects of the background elements for transfer learning purposes. The idea was to extract the relevant features of plant disease by leveraging the knowledge gained by the DL model from agricultural datasets. In this regard, the next step was weight optimization utilizing our recently presented weed-detection pipeline on these three agricultural datasets.The pipeline/framework consists of five steps (Saleem et al., 2022b): studying the effects of image resizing techniques such as fixed-shape and aspect ratio resizers; combining image interpolators such as bilinear, bicubic, area, and nearest neighbor; application of weight initializers including truncated normal, scaling variance, and random normal initializers; understanding the effects of the absence/presence of batch normalization; and use of DL optimizers such as stochastic gradient descent (SGD) with momentum, root mean square propagation (RMSProp), and adaptive moment estimation (Adam). In this step, the weights of the DL model are updated. Subsequently, the optimized weights of the best-obtained DL model were used to transfer the knowledge for the NZDLPlantdisease-v2 dataset. The mAP attained in this step was compared with that obtained in the first phase of the study. To validate the results, the same optimization pipeline was applied directly to the NZDLPlantDisease-v2 dataset using pre-trained weights on the COCO dataset. A higher AP of the healthy/disease classes established the proposed hypothesis that the best DL architecture is initially trained with a large general-purpose dataset, optimized with image resizing techniques and weight optimization algorithms through agricultural datasets, and retrained through updated weights on the proposed dataset attained better identification of plant diseases.

Finally, the effectiveness of the research was demonstrated by testing the optimized model with an externally generated dataset and using a stratified five-fold cross-validation technique. The research methodology is illustrated in Figure 1.

Proposed dataset

This research presents a new dataset called NZDL PlantDisease-v2, which contains diseases in eight prominent NZ vegetables, including tomato, potato, peas, beans, cabbage, cauliflower, broccoli, and kumara. The Samsung Galaxy S10 Plus smartphone with the following specifications was used to take the dataset images: three cameras with 12 MP f/1.5-2.4 (wide), 12 MP f/2.4 (telephoto), and 16 MP, f/2.2 (ultrawide). Dataset images were collected from various local horticultural fields in Auckland and Palmerston North, New Zealand. The working distance of the images is 200-300 mm. The data collection period was from December 2020 to May 2021.

The dataset was divided into three sub-datasets: training (70%), validation (20%), and testing (10%). An open-source tool LabelImg was used to annotate the dataset images. According to the requirements of the TensorFlow object detection API, the XML files were converted into CSV files and then transformed into TFrecords. A sample of each annotated class is shown in Figure 2A. Various real agricultural conditions, such as the absence/presence of shadows, different lighting conditions, and sudden weather changes, were considered, as shown in Figure 2B. Moreover, other practical conditions have been considered by imaging plant organs with multiple diseases. For example, black rot and ring spot disease in broccoli and cauliflower leaves were present simultaneously (Figure 2C). However, some crops contain the disease only in their leaves. The details of the proposed dataset are presented in Table 1.

Data augmentation

The size of the dataset was initially increased by cropping a group of healthy/disease classes containing more than one object of interest. Several data augmentation methods were applied, such as a 30% change (increase or decrease) in brightness, sharpness, and contrast (Wang et al., 2021a). In addition, the effects of noise, including Gaussian and Laplacian noise, were evaluated. For this purpose, open-source software named XnViewMP was used. The maximum intensities were 10.0, and 50.0, out of which random intensities of 2.0 and 10.0 were respectively selected. Furthermore, more diversity in the dataset

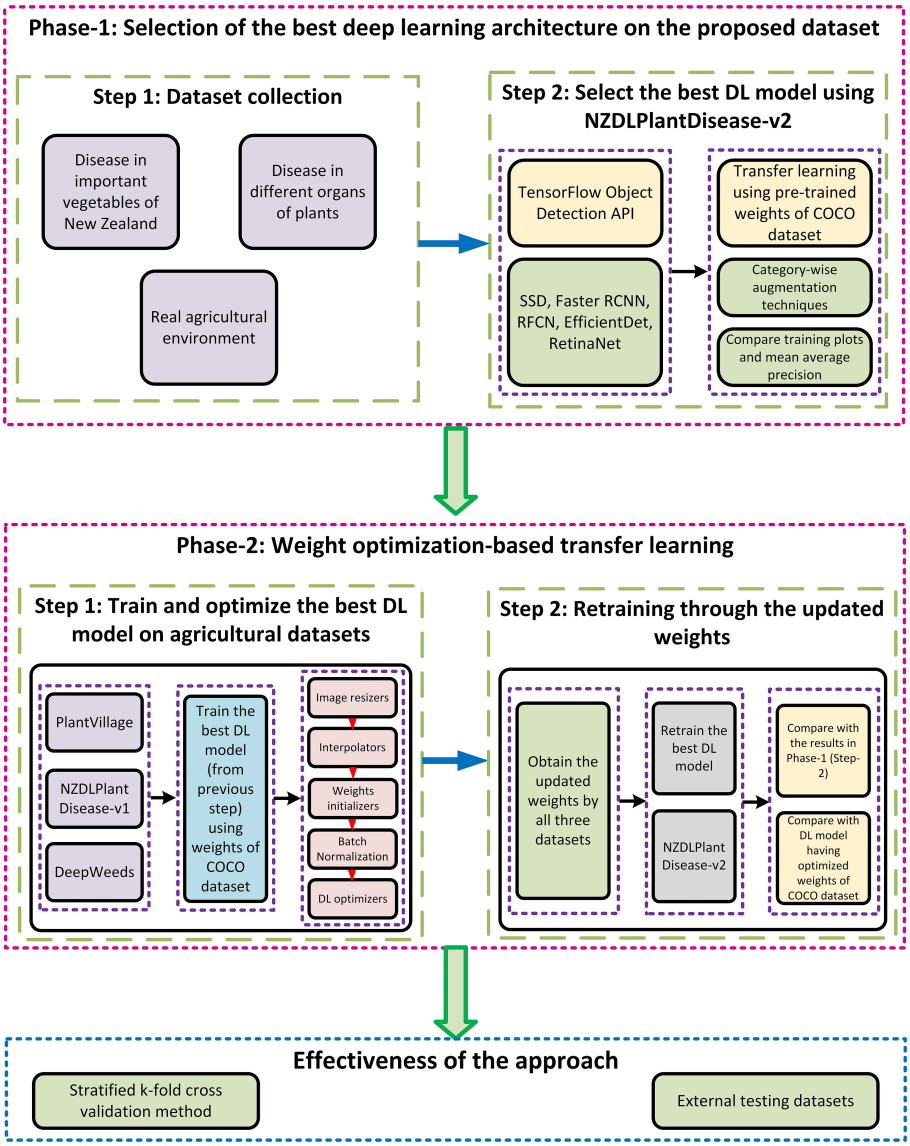


FIGURE 1
Overall workflow of the proposed methodology.

was obtained by rotational/translational changes, such as 90°, -90°, 180°, horizontal, and vertical changes. An example of augmented images for the tomato late blight class is shown in Figure 3.

For a comprehensive analysis, the data enhancement methods were arranged into five groups. The first contains only original (OO) images, then a combination of original with translational/rotational changes (OT), original with color variation (OC) (increase/decrease in brightness, contrast, and sharpness), original with noise and variation in color change simultaneously (OCN), and finally a combination of all the methods (OTCN).

Agricultural datasets

Three agricultural datasets were selected to extract the distinct features of different diseases in the plant species. The first dataset was the commonly used plant disease dataset, called PlantVillage (PV). The PV dataset contained 38 classes of healthy and diseased plant leaves from 14 plant species (Hughes and Salathé, 2015). The dataset images were generated in a laboratory-controlled environment. The former version of the proposed dataset series, NZDLPlantDisease-v1 (Saleem et al., 2022a), was selected. This dataset comprises 20 classes of healthy and defective plant leaves, stems, and fruits from New Zealand's

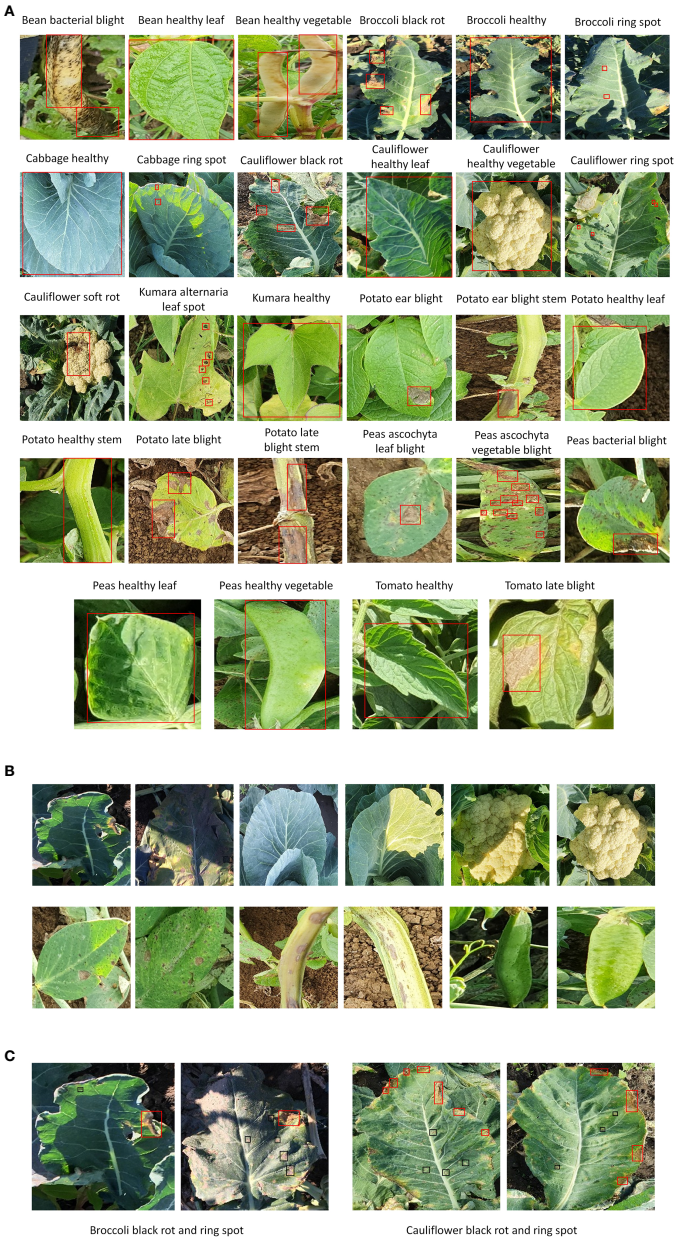


FIGURE 2
Samples of NZDLPlantDisease-v2 dataset. **(A)** Annotated example of each class. **(B)** Examples of healthy and defective organs of vegetables in different environmental conditions. **(C)** Examples of multiple diseases in plant leaves.

five important horticultural crops. All the dataset images were collected in a real horticultural environment. The third dataset is the DeepWeeds dataset, which contains eight classes of weeds and a negative/non-weed class from northern Australia (Olsen et al., 2019). The DeepWeeds dataset was also considered a real agricultural background when collecting the dataset images. An annotated example of each class from these agricultural datasets is presented in Figure 4.

Deep learning framework and models

In this study, TensorFlow object detection APIs 1 and 2 were used to train the DL models. The pre-trained weights on the Common Object in Context (COCO) dataset was used to train the DL architectures in the first phase. To increase the training speed, an NVIDIA Graphical Processing Unit GeForce GTX 1080 Ti was used with the following specifications: 11 GB

TABLE 1 Summary of NZDLPlantDisease-v2 dataset.

Crops	Plant Organs	Pathogens	NZDLPlantDisease-v2 classes with number of images	Simplified annotation labels
Bean	Leaf	–	Healthy (52)	B_healthy_l
	Vegetable	Bacteria	Bacterial blight - Xanthomonas axonopodis pv. phaseoli (93)	B_bac_blight_v
		–	Healthy (55)	B_healthy_v
Broccoli	Leaf	Bacteria	Black rot - Xanthomonas campestris pv. campestris (53)	Br_blk_rot
		Fungi	Ring spot - Mycosphaerella brassicicola (83)	Br_r_spot
		–	Healthy (74)	Br_healthy
Cabbage	Leaf	Fungi	Ring spot - Mycosphaerella brassicicola (192)	C_r_spot
		–	Healthy (173)	C_healthy
Cauliflower	Leaf	Bacteria	Black rot - Xanthomonas campestris pv. campestris (144)	Cf_blk_rot
		Fungi	Ring spot - Mycosphaerella brassicicola (48)	Cf_r_spot
		–	Healthy (68)	Cf_healthy_l
	Vegetable	Bacteria	Soft rot - Erwinia carotovora subsp. carotovora (36)	Cf_s_rot
		–	Healthy (96)	Cf_healthy_v
Kumara	Leaf	Fungi	Alternaria leaf spot - fungus Alternaria spp. (50)	K_alt_lf_spot
		–	Healthy (103)	K_healthy_l
Potato	Leaf	Fungi	Ear blight - Alternaria solani (119)	Po_ear_blight
		Oomycete	Late blight - Phytophthora infestans (57)	Po_lt_blight
		–	Healthy (199)	Po_healthy_l
	Stem	Fungi	Ear blight - Alternaria solani (144)	Po_ear_blight_s
		Oomycete	Late blight - Phytophthora infestans (89)	Po_lt_blight_s
		–	Healthy (107)	Po_healthy_s
Peas	Leaf	Fungi	Ascochyta blight - Mycosphaerella pinodes (224)	Ps_asc_blight
		Bacteria	Bacterial blight - Pseudomonas syringae pv. pisi (68)	Ps_bac_blight
		–	Healthy (115)	Ps_healthy_l
	Vegetable	Fungi	Ascochyta blight - Mycosphaerella pinodes (143)	Ps_asc_blight_v
		–	Healthy (101)	Ps_healthy_v
Tomato	Leaf	Oomycete	Late blight - Phytophthora infestans (133)	T_lt_blight
		–	Healthy (220)	T_healthy_l

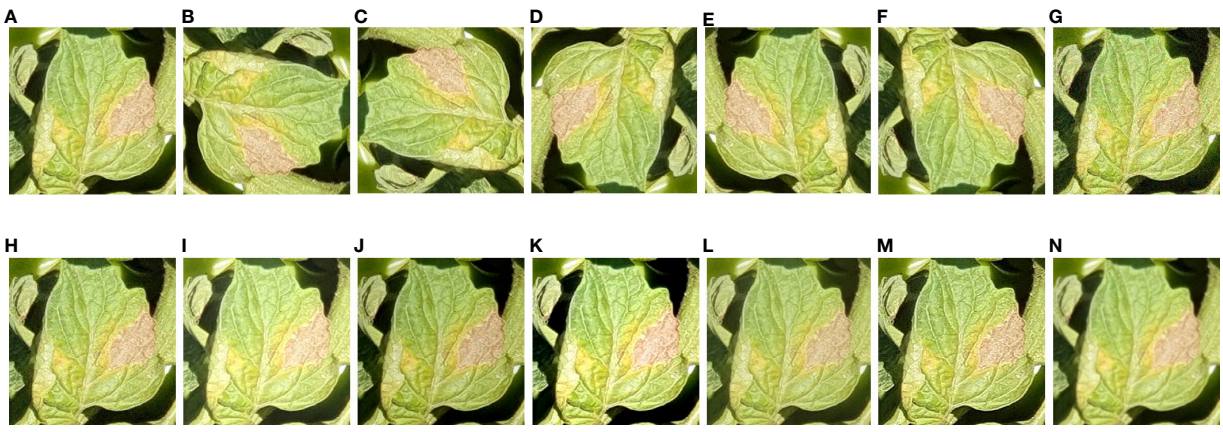


FIGURE 3
Examples of tomato late blight with all augmentation techniques (including original image). **(A)** Original. **(B)** 90°. **(C)** -90°. **(D)** 180°. **(E)** Horizontal. **(F)** Vertical. **(G)** Gaussian noise, **(H)** Laplacian noise **(I)** High brightness. **(J)** Low brightness. **(K)** high contrast. **(L)** low contrast. **(M)** high sharpness. **(N)** low sharpness.

memory, 3584 CUDA cores, 484 GB/s memory bandwidth, and a 1582 MHz boost clock. To accelerate training, the CuDNN library was imported into the system.

For this research, various state-of-the-art DL models were trained, including RetinaNet (Lin et al., 2017), EfficientDet (Tan et al., 2020), faster region-based convolutional neural network (Faster RCNN) (Ren et al., 2015), single-shot multibox detector (SSD) (Liu et al., 2016), and region-based fully convolutional network (RFCN) (Dai et al., 2016). Some models were trained using different feature extractors, as available in the TensorFlow API. The performance of the DL architectures was evaluated by mean average precision (mAP), which was used to analyze the performance of several DL models, such as Faster RCNN (Ren et al., 2015), SSD (Liu et al., 2016), and RFCN (Dai et al., 2016).

Optimization techniques

Image resizers and interpolators

The first step in the performance optimization of the best-obtained DL model was studying the effects of two image resizing techniques: aspect ratio (AR) and fixed-shape (FS) resizers. Image resizers were applied to the interpolators, including bilinear, area, bicubic, and nearest neighbor.

Weight initializers

The next step was to evaluate the three weight initialization methods. First, the truncated normal (TR) was analyzed to remove dead neurons owing to the use of the ReLU activation function. Then, scaling variance (SV) was applied, which was beneficial for maintaining the variance of the output layer with the input layers (He et al., 2015). The random normal (RN) was also tested, which helped to create tensors through normal distribution.

Batch normalization

The effects of batch normalization (BN) were also analyzed and used to solve the internal covariate shift. This neural network (NN) problem occurs because of the variation in the input of the NN distribution with the variation in the parameters of the previous layer (Ioffe and Szegedy, 2015). Furthermore, this method accelerates neural network training.

DL optimizers and hyperparameter selection

Three well-known DL optimization algorithms were used: stochastic gradient descent (SGD) with momentum (Ruder, 2016), adaptive moment estimation (Adam) (Kingma and Ba, 2014), and root mean square propagation (RMSProp) (Hinton et al., 2012). Hyperparameters were selected using the random search method (Bergstra and Bengio, 2012), such as the learning rate (LR), momentum (mom), epsilon (eps), and discounting factor (rho), as presented in Table 2.

Validation techniques

External dataset

The effectiveness of this study was validated by generating an external dataset. This dataset contains the healthy and diseased plant classes of NZDLPlantDisease-v2 using a random Google search. This was done to ensure that the proposed methodology is valid in different agricultural and environmental situations.

Cross-validation method

This method considers the class imbalance problem (the number of samples is different for each class) in the dataset for validation of the proposed method. The stratified five-fold cross-validation technique was used to maintain the original sample size of each class in each fold (He and Ma, 2013). Otherwise, a biased distribution of the dataset images could have occurred if all the samples had been randomly mixed and divided into a certain number of folds.

Results

This study aimed to detect plant diseases using a newly generated dataset named NZDLPlantDisease-v2. The proposed approach is divided into two phases, each containing two steps. The first step of the initial phase is dedicated to the collection, augmentation, and annotation of the dataset, as explained in the previous section. The subsequent steps are described in this section. Following the methodology shown in Figure 1, the results from the second step of the first phase to the second step of the final phase are presented in this section.

Phase 1: Selection of the best DL architecture and data augmentation technique using NZDLPlantDisease-v2

Training through pre-trained COCO weights

Initially, the DL architectures were trained on the original images (without augmentation) of the proposed dataset. The two best DL architectures that achieved the highest mAP were selected. Further analysis of both DL models was performed by training them using different augmentation techniques. As default configurations, models such as Faster RCNN and RFCN were trained with aspect ratio image resizing methods with a minimum of 600 and a maximum of 1000 pixels, respectively. However, in EfficientDet, the same resizer with dimensions of 512 was used. Furthermore, the RetinaNet and SSD models were trained with a fixed shape resizer with 300 × 300 pixels.

The tradeoff between accuracy and training time was addressed by testing different batch sizes; the most feasible was

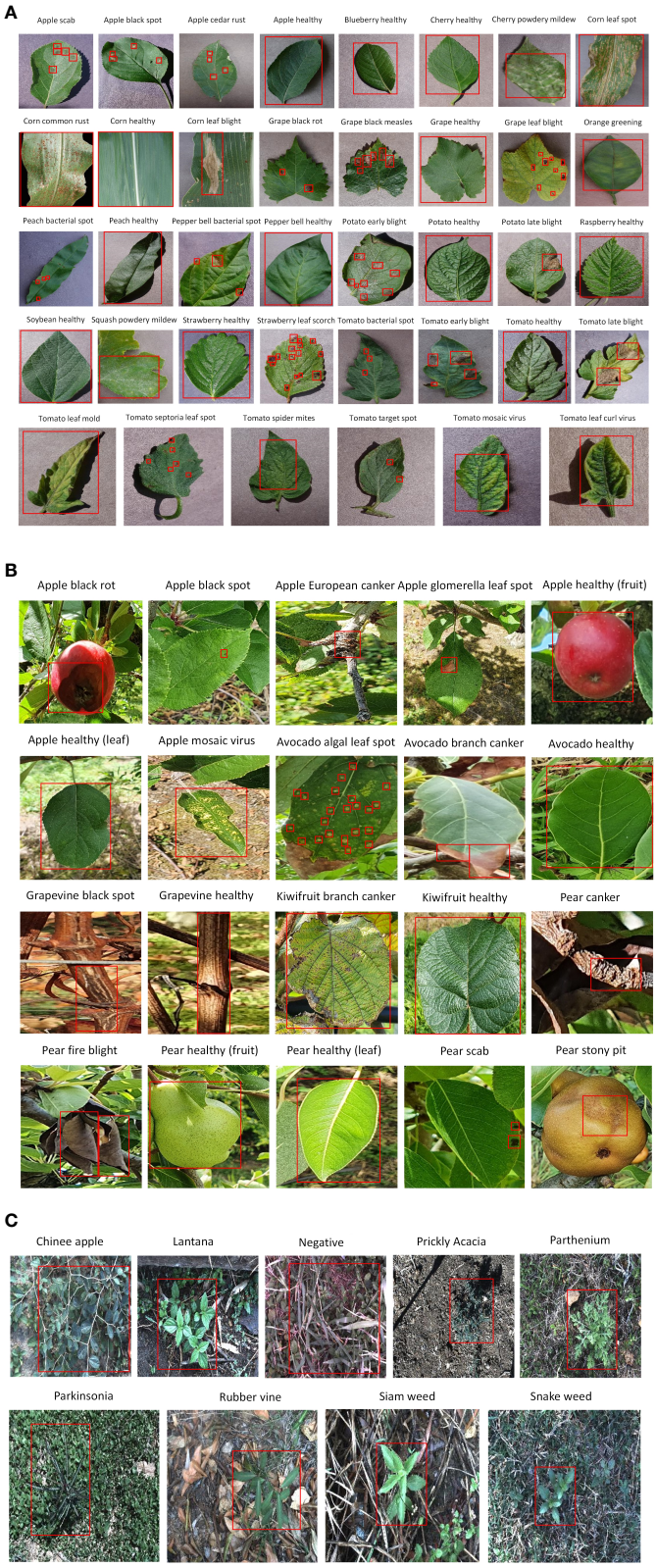


FIGURE 4
Annotated example of each class from agricultural datasets. **(A)** PlantVillage. **(B)** NZDLPlantDisease-v1. **(C)** DeepWeeds.

TABLE 2 Hyperparameters of deep learning optimizers.

DL optimizers	DL meta-architectures	Hyperparameters
SGD with momentum	EfficientDet	LR = 3 x 10-3, mom = 0.9
	Faster RCNN ResNet-50	LR = 2 x 10-4, mom = 0.9
	Faster RCNN ResNet-101	LR = 2 x 10-4, mom = 0.9
	Faster RCNN Inception-v2	LR = 1 x 10-4, mom = 0.9
	R-FCN ResNet-101	LR = 3 x 10-4, mom = 0.9
	SSD Inception-v2	LR = 3 x 10-3, mom = 0.9
	SSD MobileNet-v2	LR = 2 x 10-4, mom = 0.9
	SSD ResNet-50 (RetinaNet)	LR = 4 x 10-4, mom = 0.9
SGD with momentum	Faster RCNN Inception ResNet-v2	LR = 3 x 10-4, mom = 0.9
Adam		LR = 1 x 10-4, eps = 1.0-2
RMSProp		LR = 2.5 x 10-4, rho = 0.9, mom = 0.9, eps = 1 x 10-3

found to be 4. SGD with momentum was the default optimizer for training each model. Using empirical observations, all DL models were trained for 200k iteration steps. However, the required number of iteration steps for the training convergence is different for each model. For example, Faster RCNN Inception ResNet-v2 required the least number of iterations, approximately 110k steps. SSD Inception-v2 required the highest number of iteration steps, approximately 190k steps. However, RFCN required 120k steps, Faster RCNN ResNet-50, Faster RCNN ResNet-101, Faster RCNN Inceptionv2 required 160k iteration steps, SSD MobileNet-v2, and RetinaNet required 170k steps, whereas EfficientDet required 180k steps. In terms of the computation/training time, SSD MobileNet was the fastest model, requiring only 5.5 hours, and Faster RCNN Inception ResNet-v2 was the slowest model, which took approximately 18 hours to complete the training. The main observations from this step are as follows.

- The training and validation plots of all DL models did not show any sign of overfitting, as the losses converged and had no abrupt increase in the loss after the final iteration step.
- The training and validation profiles of the DL architecture are shown in Figure 5. The Faster RCNN ResNet-101 had the lowest training loss of approximately 0-0.1%, followed by Faster RCNN ResNet-50 and RFCN with 0-0.15% loss, Faster RCNN Inception-v2 with 0-0.2% loss, and Faster RCNN Inception ResNet-v2 with 0-0.25% loss. The rest of the DL models attained a comparatively higher training loss between 0-0.5%, 0-0.7%, 0-0.8%, and 0.5-1% for RetinaNet, SSD Inception-v2, EfficientDet, and SSD MobileNet-v2, respectively. Similarly, the lowest validation loss was obtained by Faster RCNN ResNet-101 followed by RFCN ResNet-101, Faster RCNN ResNet-50, and Faster RCNN Inception ResNet-v2.
- In terms of mAP, Faster RCNN Inception ResNet-v2 achieved the highest value of 63.74%, followed by Faster RCNN ResNet-101 with 60.34%, as shown in Table 3. This was due to the high (>80%) average precision (> 80%) of 11 and 10 classes by Faster RCNN trained with Inception ResNet-v2 and ResNet-101 feature extractors, respectively. The selection of the two best DL models was validated by retraining each model using all augmentation techniques. Similar models achieved the best results; for example, Faster RCNN Inception ResNet-v2 achieved 47.83% and Faster RCNN ResNet-101 achieved 45.25% mAP.
- EfficientDet and RetinaNet achieved the lowest mAP owing to several undetected classes, as indicated by the zero AP in Table 3.
- It was also observed that few classes were successfully detected with Faster RCNN Inception ResNet-v2 that were undetected (false negative) or wrongly identified (false positive) with other DL models, such as broccoli ring spot (Br_r_spot) and cabbage healthy (C_healthy), as shown in (Figures 6A, B). Similarly, potato early blight attained the highest average precision with the best-selected DL models (Figure 6C). However, SSD MobileNet and EfficientDet were unable to detect some of the early blight images of potatoes (Figure 6D).
- Some of the classes performed well in all or most of the DL models, including bean bacterial blight, healthy cauliflower (vegetable), healthy cabbage, healthy potato stem, and healthy potato leaves.
- Although this step yielded the two best DL models, the mAP was not satisfactory. There was a significant margin in the performance improvement. This was because a few classes could not be detected, including the broccoli black rot, cauliflower ring spot, kumara Alternaria leaf spot, and pea bacterial blight, by any of the DL architectures. Several examples of the false

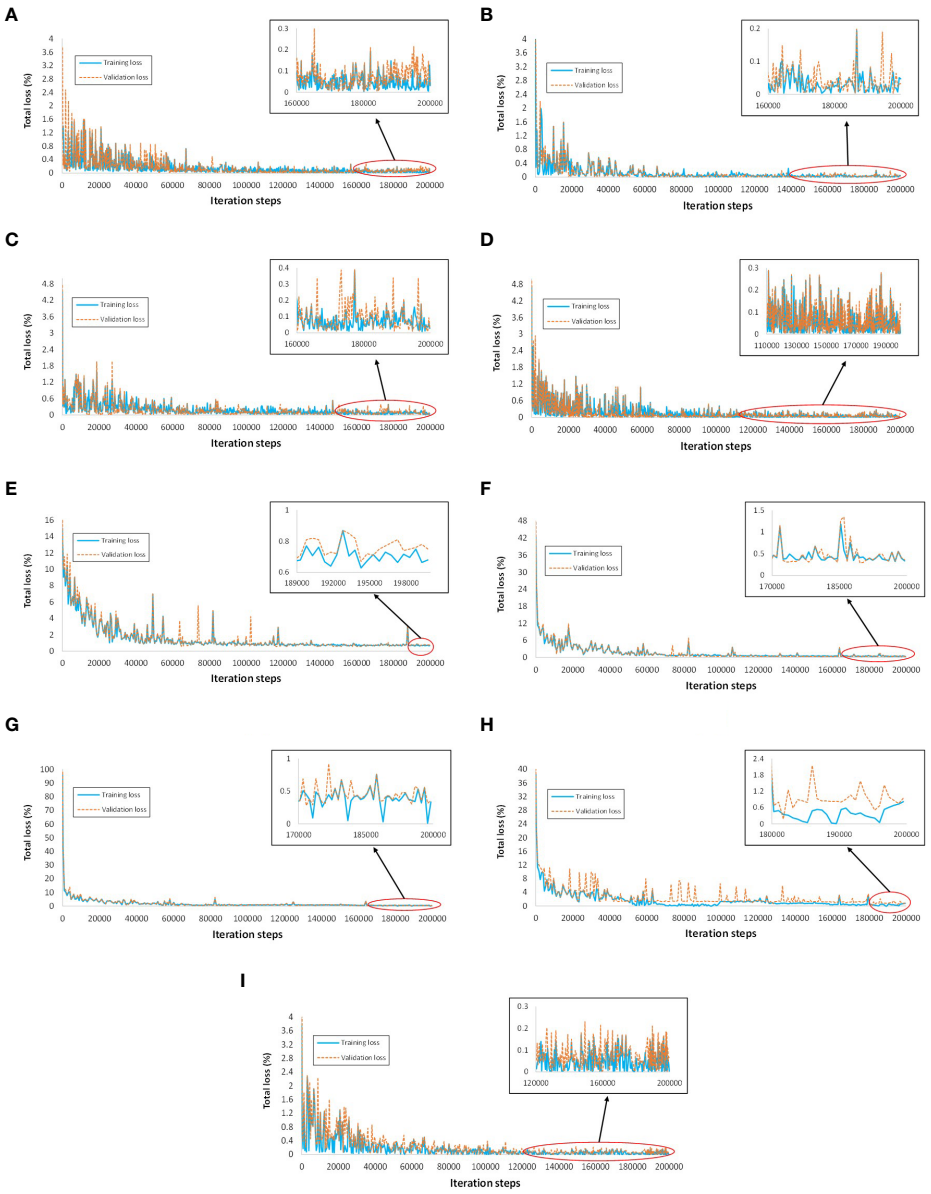


FIGURE 5
Training and validation loss plots of various DL architectures. **(A)** Faster RCNN ResNet-50. **(B)** Faster RCNN ResNet-101, **(C)** Faster RCNN Inception-v2. **(D)** Faster RCNN Inception ResNet-v2. **(E)** SSD Inception-v2. **(F)** SSD MobileNet-v2. **(G)** RetinaNet. **(H)** EfficientDet. **(I)** RFCN ResNet-101.

positive and false negative results of the different DL models are presented in Figures 6E–H.

- After considering all data augmentation techniques, some of the classes improved their AP, such as healthy broccoli, healthy beans (leaves and vegetables), and cabbage ring spots. However, the overall performance (mAP) of the Faster RCNN Inception ResNet-v2 was degraded due to the small AP of several classes. These classes included healthy cauliflower (leaves and vegetables), kumara leaves, healthy potato early blight (leaves and stems), potato (leaves and stems), potato blight, Ascochyta blight of peas, and healthy pea leaves.

Category-wise study of data augmentation techniques

To understand the effects of the data augmentation techniques on the DL model, we divided them into five categories, as explained in the previous section. The two best

TABLE 3 Performance of deep learning meta-architectures in terms of average precision (%) of each class.

Classes of NZDLPlantDisease-v2	Deep learning architectures with feature extractors								
	Faster RCNN				SSD			EfficientDet	RFCN
	ResNet-50	ResNet-101	Inception-v2	Inception-ResNet-v2	Inception-v2	MobileNet-v2	ResNet-50 (RetinaNet)	EfficientNet	ResNet-101
Br_blk_rot	3.84	2.59	3.38	13.64	0	0	0	0	3.06
Br_healthy	29.24	36.36	18.18	36.82	45.45	43.94	38.55	23.81	21.97
Br_r_spot	23.04	41.68	33.08	72.74	6.44	12.88	0	0	34.91
B_bac_blight_v	89.61	89.61	96.16	97.49	72.73	62.63	43.10	31.49	80.42
B_healthy_l	33.33	36.36	36.82	33.33	34.55	21.97	0	0	36.36
B_healthy_v	27.27	45.45	27.27	49.91	9.09	21.97	11.49	42.10	59.69
Cf_blk_rot	55.69	59.66	49.19	58.61	22.58	20.43	8.29	53.18	61.08
Cf_healthy_l	31.19	30.5	30.83	54.7	43.01	45.45	0	0	16.41
Cf_healthy_v	90.91	90.91	81.82	90.91	81.82	90.91	59.45	65.99	90.91
Cf_r_spot	17.77	4.85	0.91	14.77	0	0	0	0	3.62
Cf_s_rot	49.57	63.64	70.99	36.36	49.55	45.45	0	11.25	60.29
C_healthy	50.69	81.21	58.56	94.54	80.56	80.68	0	10.05	69.43
C_r_spot	66.67	98.61	72.03	96.72	22.98	23.03	35.57	30.13	100
K_healthy_l	81.82	80.17	90.08	81.82	56.01	80.91	42.98	27.65	81.82
K_alt_lf_spot	9.09	23.48	0	37.02	0	0	0	0	28.83
Po_lt_blight_s	57.52	36.01	28.44	32.55	34.22	32.83	0	0	51.27
Po_ear_blight	71.08	86.48	72.44	88.45	59.87	54.17	60.02	43.66	67.87
Po_ear_blight_s	42.17	59	67.65	69.19	24.84	23.97	21.33	28.18	57.85
Po_healthy_l	88.76	86.56	90.91	78.1	70.69	79.44	19.74	38.59	96.28
Po_healthy_s	90.08	90.91	80.17	89.39	72.73	72.73	77.73	53.88	88.16
Po_lt_blight	36.58	51.23	57.56	52.99	0	4.55	0	0	44.21
Ps_asc_blight	72.24	79.98	78.97	71.43	54.21	35.86	0	10.25	79.67
Ps_asc_blight_v	89.26	79.17	87	87.63	100	72.73	46.52	4.83	81.82
Ps_bac_blight	18.08	31.47	26.71	14.16	4.55	6.93	0	0	41
Ps_healthy_l	100	98.6	97.78	90.91	100	89.17	63.38	74.12	89.4
Ps_healthy_v	90.91	81.82	90.91	90.91	90.91	79.9	0	25.88	81.82
T_healthy_l	72.73	72.73	81.82	81.82	72.73	81.82	37.81	35.91	72.73
T_lt_blight	42.93	50.44	37.13	67.74	35.45	27.27	0	0	63.49
mAP (%)	54.72	60.34	55.96	63.74	44.46	43.27	20.21	21.82	59.44

DL architectures were retrained for each category. The results are discussed below.

- Among the five augmentation categories, OT achieved the best results in terms of the highest mean average precision. The performance of the Faster RCNN Inception ResNet-v2 model was significantly improved as compared to the non-augmented categories, with a difference of 18.34% in mAP.
- Nine classes were prominently detected after the application of the OT category. These classes included healthy broccoli, healthy beans, healthy cauliflower black rot, healthy cauliflower (leaves), soft cauliflower rot, kumara Alternaria leaf spot, potato late blight (stem) potato early blight (stem), and Ascochyta blight, which improved the individual AP by 63.18%, 66.26%, 17.15%, 43.04%, 63.18%, 61.91%, 48.38%, 25.59, and 27.55%, respectively.
- The OT category also attained the best performance for Faster RCNN ResNet-101. The mAP improved by approximately 10.49% compared to the results obtained with original/non-augmented images.
- It was further noticed that OTCN and OO achieved the lowest mAP for both DL models, followed by OC and OCN categories.
- The change in color/noise inclusion affected the detection performance, as shown in Figure 7A. When the image of a healthy broccoli leaf was tested, the background element was wrongly identified as broccoli black rot disease, and

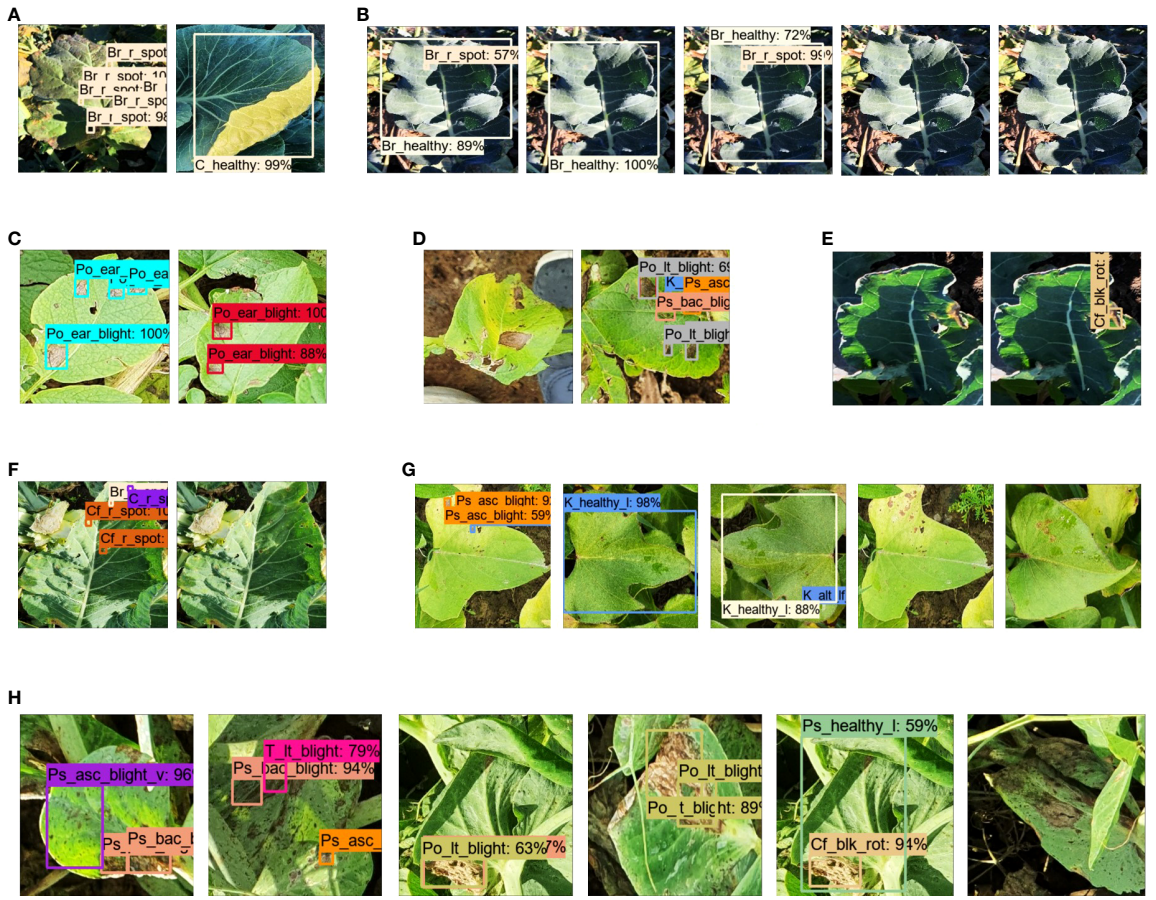


FIGURE 6
Detection results for the first step. **(A)** True positive by Faster RCNN Inception ResNet-v2 for broccoli ring spot and healthy cabbage (leaves). **(B)** False positive for broccoli ring spot by Faster RCNN ResNet-50, SSD Inception, SSD MobileNet, and the last two are undetected (false negative) by RetinaNet, EfficientDet for broccoli ring spot. **(C)** True positive by Faster RCNN Inception ResNet-v2 and ResNet-101 for potato early blight. **(D)** False negative and false positive by SSD MobileNet and EfficientDet, respectively for potato early blight. **(E)** False negative and false positive by Faster RCNN ResNet-101 and Inception ResNet-v2, respectively for broccoli black rot. **(F)** False positive by Faster RCNN ResNet-50 and false negative by SSD Inception-v2 for cauliflower ring spot. **(G)** False positive by EfficientNet, RFCN ResNet-101, Faster RCNN Inception ResNet-v2, and false negative by SSD MobileNet, RetinaNet for kumara Alternaria leaf spot. **(H)** False positive by Faster RCNN ResNet-50, ResNet-101, Inception-v2, SSD with Inception-v2, and MobileNet-v2; false negative by EfficientDet for peas bacterial blight.

the blackness in the leaf was recognized as pea Ascochyta blight. Some other examples of true positive and false positive results with the respective augmentation categories are presented in Figures 7B–E.

- Although this step of the study improved the performance of the Faster RCNN Inception ResNet-v2, few classes achieved a low AP <50%, such as broccoli black rot, healthy beans (vegetables), cauliflower ring spot, potato late blight, and peas bacterial blight.

Phase 2: Performance optimization by weights obtained from agricultural datasets

This phase of the research was divided into two steps: obtaining the optimized weights on agricultural datasets using different techniques and using the knowledge/features extracted from the agricultural datasets to the proposed dataset.

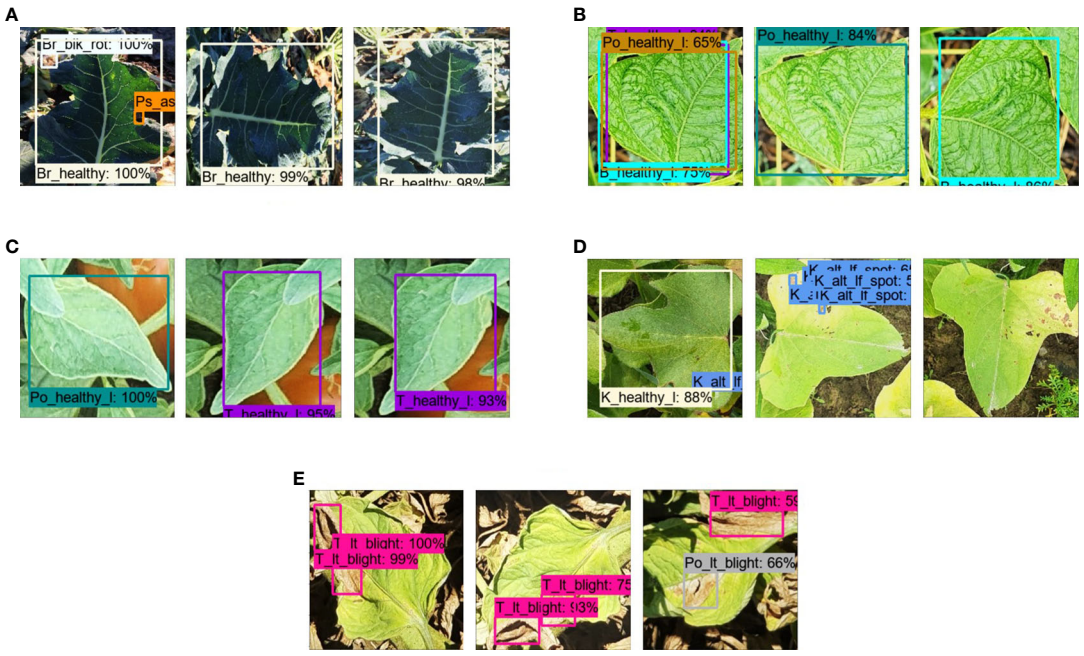


FIGURE 7
Plant disease detection by different augmentation techniques. **(A)** False positive by OC and true positive by OT and OTCN. **(B)** False positive by OO, OC, and true positive by OT for healthy beans (leaves). **(C)** False positive by OC and true positive by OT and OCN for healthy tomato leaves. **(D)** False positive by OO, true positive by OT, and false negative by OTCN for kumara Alternaria leaf spot. **(E)** True positive by OT and OCN, and false positive by OC for tomato late blight.

Effects of image resizers, interpolators, weight initializers, batch normalization, and DL optimizers on agricultural datasets

First, the best-obtained DL model, Faster RCNN Inception ResNet-v2, was trained on each agricultural dataset using default settings. Subsequently, various image resizers, interpolators, weight initializers, batch normalization, and DL optimizers are applied to determine the most suitable combination. The weights obtained through the best-performing configuration in terms of the highest mAP and highest number of detected classes (for each dataset) were saved for reuse in the later stage of this phase of the research. A summary of the results by default and optimized settings is presented in Table 4.

Optimization on the PlantVillage dataset

After training the Faster RCNN Inception ResNet-v2 with default settings, the effects of the aspect ratio and fixed shape resizer were studied on the PlantVillage dataset. Image resizers were evaluated along with interpolators to obtain the best-suited arrangement. In this regard, an aspect ratio resizer with a bilinear interpolator achieved an mAP of 61.11%. It was improved by 3.17% by training through a fixed-shape resizer trained with a bilinear interpolator with a pixel size of 300 × 300 pixels. The reason for this was the true positive detection results in the classes including apple scab, healthy peach, healthy soybean, and squash powdery mildew. Furthermore, the model performed well after adding pixels by taking the average of the color values of the neighboring pixels by bilinear interpolation.

Similarly, a performance enhancement of 4.95% in the mAP was attained after the application of a random normal initializer. Therefore, the Faster RCNN model initialized with tensors having a normal distribution was found to be the best method for initializing the weights of the neural network. Next, batch normalization was applied with different values of epsilon and decay. It was empirically found that a 0.9 value for decay and an epsilon of 0.01 were the most suitable values. However, the mAP did not improve and was 65.68%. Therefore, the next step was applied in the absence of batch normalization. The last step was to study the effects of DL optimizers on the Faster RCNN model for the PlantVillage (PV) dataset. The Adam optimizer could not extract the features of several classes and achieved a low mAP of 26.53%. However, RMSProp achieved an mAP of 61.695%, which was not an improvement over the default (SGD with momentum) optimizer. Therefore, the final settings for the PV dataset were the use of a fixed-shape resizer with a bilinear interpolator, random normal initializer, SGD with momentum optimizer, and no batch normalization, as presented in Table 4.

TABLE 4 Summary of each step of the proposed methodology.

Steps of the proposed approach		Training Specifications (Image resizers, interpolators, initializer, batch normalization, and DL optimizers)	Training and testing performance					Analysis and remarks
			DL models	Training time (h)	Training loss (%)	Validation loss (%)	mAP (%)	
Obtain the best-suited DL architecture		AR with bilinear TN, SV No BN SGD	Faster RCNN Inception ResNet-v2	18	0.13	0.35	63.74	Two of the best DL models achieved the highest mAP; some of the classes were not detected.
			Faster RCNN ResNet-101	12	0.06	0.2	60.34	
Application of data augmentation techniques		Original and translation/rotational changes Default training settings	Faster RCNN Inception ResNet-v2	18	0.425	0.61	82.08	Both DL architectures achieved better results after adding translational/rotational images to the training dataset.
			Faster RCNN ResNet-101	12	0.375	0.52	70.83	
Training on agricultural datasets	PV	Default settings	Faster RCNN Inception ResNet-v2	18	0.54	0.75	61.11	The Faster RCNN Inception ResNet-v2 was trained on different datasets with default settings.
	NZDL-1		Faster RCNN Inception ResNet-v2	18	0.42	0.62	71.48	
	DW		Faster RCNN Inception ResNet-v2	18	0.63	0.75	80.81	
Weight optimization on agricultural datasets	PV	FS with bilinear, RN, without BN, SGD	Faster RCNN Inception ResNet-v2	12	0.46	0.64	69.23	Different performance enhancement techniques were applied to the agricultural datasets to obtain optimized weights; the best combination of image resizer/interpolator, initializer, batch normalization, and DL optimizers was found to depend on the selected datasets.
	NZDL-1	FS with bicubic, RN, BN, SGD	Faster RCNN Inception ResNet-v2	12	0.31	0.46	73.32	
	DW	FS with bilinear, TR, SV, without BN, RMSProp	Faster RCNN Inception ResNet-v2	12	0.49	0.66	84.36	
Retrain through the best-optimized		PV dataset: FS with bilinear, RN, without BN, SGD	Faster RCNN	12	0.34	0.50	91.33	Used optimized weights to train Faster RCNN Inception ResNet-

(Continued)

TABLE 4 Continued

Steps of the proposed approach	Training Specifications (Image resizers, interpolators, initializer, batch normalization, and DL optimizers)	Training and testing performance					Analysis and remarks
		DL models	Training time (h)	Training loss (%)	Validation loss (%)	mAP (%)	
weights on NZDLPlantDisease-v2		Inception ResNet-v2					v2 on NZDLPlantDisease-v2 dataset; achieved a significantly higher mAP.
Compare the results with optimized weights on the COCO dataset	FS with bilinear, TR, SV, without BN, SGD	Faster RCNN Inception ResNet-v2	12	0.39	0.56	85.85	Another comparison was made with the optimized weights on the COCO dataset.
Validation of the approach	External dataset	–	–	–		93.20	Most of the classes attained a high mAP with the optimized weights.
	Stratified k-fold cross-validation	–				88.53	No noticeable differences in the performance of the final model were found; few classes can be considered in future studies.

Optimization on the NZDLPlantDisease-v1 dataset

Like the PV dataset, the weights of the Faster RCNN model were optimized using the NZDLPlantDisease-v1 dataset with the same framework/steps. The default configuration reached 71.48% mAP. However, an improvement of 0.28% in the mAP was observed when the input images were resized using the fixed shape method with a bicubic interpolator. A few classes, such as apple mosaic virus and healthy pear (leaves), achieved 17.01% and 50.49%, respectively, higher AP with fixed shape (bicubic) compared to aspect ratio (bilinear). However, only one class (pear scab) achieved better output with the default resizer/interpolator combination. Faster RCNN Inception ResNet-v2 also performed well for the NZDLPlantDisease-v1 dataset with a random normal initializer, and an enhancement of 1.1% mAP was observed. There was a small difference in the AP of the healthy/disease classes compared with the default initializer. Batch normalization with epsilon and decay was applied at 0.5 and 0.01, respectively. The mAP improved from 72.86% to 73.32%. Again, SGD with momentum performed well compared to the other two DL optimization algorithms.

Optimization on the DeepWeeds dataset

The last selected dataset is related to other agricultural application (weed identification). It also started with an evaluation using the default settings. Like the other two datasets, the fixed-shape resizer worked well for this dataset. The bilinear interpolation method achieved the highest mAP among all interpolators and achieved 84.20% mAP compared to 80.81% with the default resizer. However, in contrast to the previous datasets, DeepWeeds performed better on truncated normal and scaling variance initializers than on random normal initializers. Thus, it can be said that a truncated normal distribution and adaptation of the scale to the shape of weights resulted in a better initialization for weed detection. However, batch normalization did not work for this dataset, and a small performance degradation was observed, with a difference of almost 0.3%. When evaluating the DL optimizers, RMSProp attained the best outcomes and slightly improved the mAP to 84.36%.

Transfer learning by optimized weights

The last step of this research evaluates the Faster RCNN Inception ResNet-v2 model on the proposed dataset using optimized weights from each selected agricultural dataset. In this regard, the training and testing performances are analyzed, and the effects of updated/new weights are studied. The significance of this step is presented by comparing the results obtained in the previous phase (step 2) and optimizing the weights using a large dataset (COCO). Some important observations from this research are presented below:

- The mAP of the Faster RCNN model was increased to 91.33% by optimizing the weights through the PV dataset and improved by a margin of 9.25% in mAP. The total training loss was lower than 0.34% owing to a

considerable reduction of 0.12 and 0.16% in region proposal network (RPN) localization and box classifier classification losses, respectively. Moreover, a small reduction in box classifier localization and RPN objectness loss was observed at 0.05 and 0.01%, respectively. Furthermore, the validation loss was reduced to 0.5%. Almost all nine classes improved their detection outcomes and attained > 80% AP. These classes include broccoli black rot, broccoli ring spot, bean healthy (vegetable), cauliflower black rot, cauliflower ring spot, cabbage ring spot, kumara healthy, potato late blight (leaves), and pea bacterial blight. An example of each class is shown in Figure 8.

- In Table 4, the mAP with optimized weights by NZDLPlantDisease-v1 (former version in the series of the proposed plant disease dataset) was 87.68% and therefore increased by 5.6% compared to the previous step. There are several reasons for this improvement, as observed by the training and testing profiles of the model. First, the total training loss was reduced to approximately 0.36%, which was 0.425% when the model was trained using COCO weights. This reduction in the total loss was due to a small decrease in RPN objectness and localization loss of 0.01% and 0.055%, respectively. These losses produce a substantial improvement in the mAP. However, no noticeable changes were observed in the box classifier losses. Second, several classes were detected, or their individual AP was improved, such as broccoli ring spot, cauliflower black rot, cabbage ring spot, healthy kumara, and pea bacterial blight, by 30.58%, 23.34%, 23.75%, 28.94%, and 24.56%, respectively.
- However, the performance of the Faster RCNN model was not improved by using the weights obtained on the DeepWeeds dataset. The mAP was 80.73% with performance degradation in the classes such as broccoli healthy, cauliflower healthy (leaf), and potato early blight (stem).
- Finally, a similar weight optimization framework was applied to the new dataset using pre-trained weights on a large dataset (COCO). The best configurations were a fixed image resizer with a bilinear interpolator, truncated normal and scaling variance initializers without batch normalization, and an SGD optimizer. It was found that the Faster RCNN Inception ResNet-v2 attained 85.85% mAP, which was improved by 3.77% compared to the previous phase, but 5.48% and 1.83% lower than the mAP obtained through the weights of the PV dataset and NZDLPlantDisease-v1, respectively.

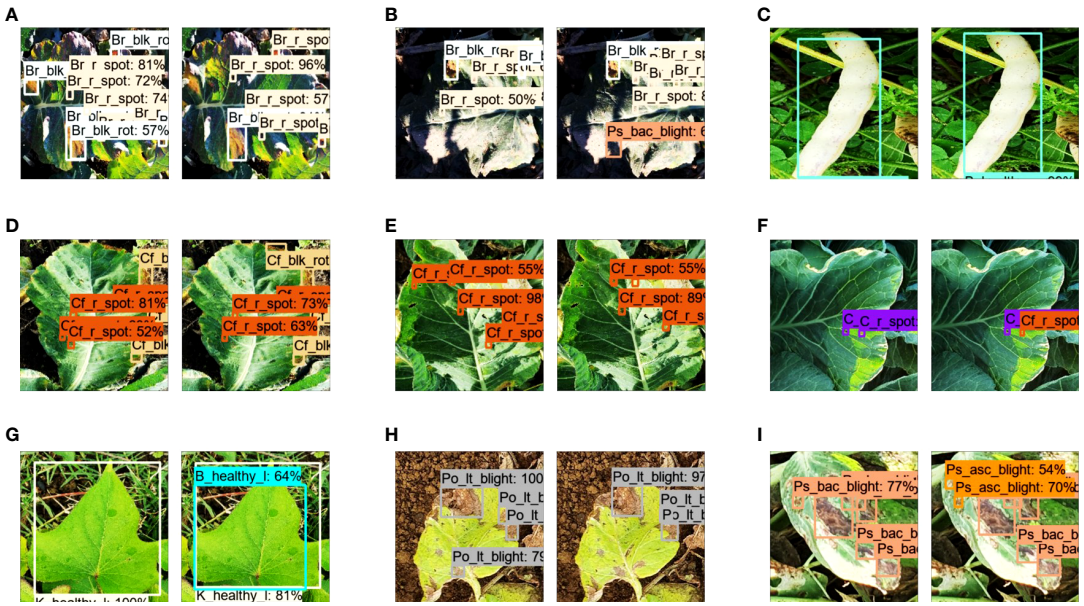


FIGURE 8
Comparison of various healthy and disease classes by optimized final model and results obtained in the previous step. **(A)** Broccoli black rot. **(B)** Broccoli ring spot. **(C)** Bean healthy vegetable. **(D)** Cauliflower black rot. **(E)** Cauliflower ring spot. **(F)** Cabbage ring spot. **(G)** Kumara healthy. **(H)** Potato late blight. **(I)** Peas bacterial blight.

Discussion

In this section, a comprehensive analysis is presented to explain the substantial improvement in the mAP using the proposed methodology for the detection of plant disease. Finally, the effectiveness of the approach is highlighted using two methods, along with describing some limitations of the study.

Analysis of the results obtained in the first and second phases of the research

- The first phase of the research was dedicated to finding the best-suitable DL model trained on the proposed dataset using pre-trained weights on a large COCO dataset along with the best data augmentation method. The main finding of this step was that the Faster RCNN Inception ResNet-v2 achieved the best results in terms of the highest mean average precision in the absence and presence of augmented images. However, some classes achieved a low AP. Interestingly, several classes performed worse when the model was trained using all augmentation (OTCN) techniques. Therefore, different data augmentation categories were investigated before attempting any improvement in the performance of the DL model.
- The Faster RCNN model attained the best results when it was trained with the translational/rotational (OT) data augmentation method. There could be several reasons for obtaining unsatisfactory results in different categories, except for the OT. First, the number of training images for the OO category was insufficient to extract the distinct features of the healthy and disease classes. After a change in the brightness, contrast, and sharpness of the images, the symptoms of the plant disease might become similar to other healthy/diseased classes of the same plant species or another, resulting in false positive outcomes. Furthermore, the features of plant diseases resemble the background elements after the change in color or the inclusion of noise in the input images as shown in Figure 7A (Barbedo, 2018). Therefore, the Faster RCNN model might not be able to extract specific features or symptoms of plant diseases.
- Furthermore, OT augmentation has practical importance. This is because the location/position of the disease spot and plant organs are different in real agricultural fields. Therefore, if the DL model performs well on translated/rotated images, it can be used in a complex agricultural environment.
- The second phase aimed to study the effects of several performance optimization methods on the agricultural datasets to obtain the optimized weights of the Faster RCNN model. Finally, the optimized weights from these agricultural datasets were tested on the proposed dataset.
- It can be observed from Table 4 that each of the selected datasets required different specifications. The combination of various techniques depended on the dataset because all three datasets had several fundamental dissimilarities, for example, background environment, number of dataset images, number of classes, and image quality. In addition, the optimized weights produced a significant improvement in the performance of Faster RCNN Inception ResNet-v2 for the three datasets. This demonstrates the importance of the performance optimization pipeline (Saleem et al., 2022b).
- From the results, it can be observed that the performance of the Faster RCNN model was improved using weights optimized by other plant disease datasets. The weights from the weed datasets did not generate any significant difference in the performance of the model.
- There are several ways to understand the reasons for obtaining an ample improvement in the Faster RCNN performance by weight obtained through the PV dataset. First, PV has a greater variety of plant disease types. Therefore, various symptoms/spots were extracted for transfer learning. Next, the PV dataset had a greater number of total samples that helped learn the distinct features successfully. Hence, the useful information through the PV dataset was more suitable as compared to the pre-trained weights of the Faster RCNN by NZDLPlantDisease-v1.
- The problems highlighted in the Introduction have been addressed. This was because the final model successfully detected all the disease classes in different plant organs of various vegetables with multiple plant diseases.
- On top of that, similar performance optimization methods were applied to the proposed dataset using the pre-trained weights of the COCO/large general-purpose dataset to get the optimized weights. Then, the Faster RCNN Inception ResNet-v2 model was retrained on the proposed dataset using the new/updated weights. The results show that leveraging the knowledge learned from the agricultural dataset performed better than leveraging from the large general-purpose dataset.

Validation of the research

The performance of the final model was validated using two methods. First, a stratified cross-validation method was used because of the class-imbalance problem. This method allows for an unbiased distribution of class samples among all folds of a dataset. The testing dataset was folded five times. The first fold was assumed to be the default testing dataset, which attained 91.33%, as explained in the previous section. The mAPs obtained from fold2, fold3, fold4, and fold5 were 91.57, 90.82, 91.14, and 91.17%, respectively. Therefore, the mAP varied with a small difference of 0.16 to 0.51%.

Subsequently, another validation was performed by testing the final model on an external dataset (generated through a random Google search). Overall, 14 classes achieved an AP of more than 90%, including broccoli black rot, broccoli healthy, bean bacterial blight, cauliflower black rot, cauliflower healthy (leaves and vegetables), potato early blight (leaves and stems), potato late blight (stem), peas Ascochyta blight (leaves and vegetables), peas healthy (vegetables), tomato healthy, and tomato late blight. However, some classes achieved an AP of less than 83%, such as cauliflower ring spot, kumara Alternaria leaf spot, kumara healthy, and potato healthy (stem). These classes should be further studied in future research. Figure 9 presents a few samples of the true and false positive outcomes from the external dataset.

Limitations of the study

The performance of the Faster RCNN model was more significantly improved by using the weights optimized through PV dataset compared to NZDLPlantDisease-v1 dataset. Therefore, it is concluded that a dataset that contains a higher number of plant disease classes can contribute to improving outcomes more effectively. The proposed hypotheses could be further explored in future studies by considering at least one more plant disease dataset in both real and laboratory environments. In this way, it can also be established that the environment of the dataset (from where optimized weights are obtained) also affects the performance of the DL model.

FIGURE 9
Results with the external testing dataset. **(A)** True positive outcomes. **(B)** False positive and false negative results.

The NZDLPlantDisease-v2 dataset contains healthy/disease classes for various organs of plant species, including beans, peas, potatoes, and cauliflower. However, some crops only contain the disease in their leaves, such as broccoli, cabbage, kumara, and tomato. Therefore, data collection should be continued with other plant organs of these vegetables. Moreover, due to time constraints and weather conditions, a few other important New Zealand vegetables, such as onion, squash, and sweet corn, could not be considered.

The problem of multiclass plant diseases in a single organ at a time has been addressed for only four classes. Hence, this problem has been analyzed in a limited manner. Moreover, the data annotation method is a lengthy process for a large number of classes. The more difficult part is to annotate the images with the multiclass problem, as each image must be labeled multiple times (for each disease). Therefore, if a plant organ suffers from a greater number of diseases, annotation time increases significantly which can lead to reducing the accuracy of data annotation. Hence, DL-based research still requires human effort before training/evaluating models to perform delicate tasks such as plant disease detection.

Conclusions and future recommendations

This research has addressed an important agricultural problem of identifying plant diseases in New Zealand vegetables by deep learning. In this regard, a new dataset named NZDLPlantDisease-v2 was generated, which contains 28 healthy/disease classes in eight plant species. This dataset also contains complex agricultural challenges, including multiple diseases in a single vegetable plant organ, diseases in different plant organs, and variations in real agricultural environments.

A two-step transfer learning approach based on weight optimization using different techniques on agricultural datasets was proposed. In the first phase, the best-suited DL architecture was found to be Faster RCNN Inception ResNet-v2 which attained the highest mAP using pre-trained weights on the COCO dataset. Then, the Faster RCNN model with translated/rotated data augmentation techniques improved the performance of the DL model, with an mAP of 82.08%. In the second phase, the knowledge/weights extracted from the agricultural dataset were transferred to learn the features of the classes of the proposed dataset. In this regard, three agricultural datasets were considered: two plant disease datasets in a controlled and real environment and a weed dataset in complex agricultural conditions. The weights obtained using the PlantVillage dataset significantly improved the performance of the Faster RCNN model. The optimization of weights was obtained having the best configurations consisting of a fixed-shape resizer along with a bilinear interpolator, random normal initializer, without batch normalization, and the SGD optimizer. The final mAP was 91.33%, and each class attained more than 80% AP. The optimized DL model outperformed the results obtained in the first step and the optimized weights using a general-purpose dataset (COCO). The effectiveness of the proposed approach was validated using a stratified k-fold cross-validation method and external testing dataset.

Overall, this research has demonstrated an extended strength of deep transfer learning for plant disease detection. The proposed methodology can be used in other agricultural applications. To extend this work, various concepts can be explored to visualize the feature extraction of healthy/diseased plant organs, such as t-distributed stochastic neighbor embedding (t-SNE) plots. Moreover, a modified DL model should be proposed to reduce the training/computation time. Furthermore, a similar approach can be tested for segmentation tasks using the latest DL architectures, such as DeepLab-v3, U-Net, feature pyramid network (FPN), and pyramid scene parsing network (PSPNet). A more in-depth analysis can be performed using other performance metrics (Wu and Zhou, 2017) for the multiclass plant disease detection problem.

Data availability statement

The dataset presented in this study is made publicly available in a GitHub repository: https://github.com/kmarif/NZDLPlantDisease-v2.

Author contributions

MHS and KMA designed the research and proposed the methodology. MHS curated the dataset, performed the experiments, and wrote the original draft. KMA reviewed and edited the manuscript. JP and KMA obtained funding and supervised this research. All authors contributed to the article and approved the submitted version.

Funding

This research was funded by the Ministry of Business, Innovation, and Employment (MBIE), New Zealand, Science for Technological Innovation (SfTI) National Science Challenge.

Acknowledgments

The authors are grateful to Dr. Carl Mesarich (School of Agriculture and Environment, Massey University) for providing the common and scientific names for plant diseases. The authors

also thank Fakhia Hammad and Muhammad Taha for their help with the annotation of the dataset images.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Abbas, A., Jain, S., Gour, M., and Vankudothu, S. (2021). Tomato plant disease detection using transfer learning with c-GAN synthetic images. *Comput. Electron. Agric.* 187, 106279. doi: 10.1016/j.compag.2021.106279

Aitken, A. G., and Warrington, I. J. (2020). Fresh facts. Auckland: Plant & Food Research.

Argüeso, D., Picon, A., Irusta, U., Medela, A., San-Emeterio, M. G., Bereciartua, A., et al. (2020). Few-shot learning approach for plant disease classification using images taken in the field. *Comput. Electron. Agric.* 175, 105542. doi: 10.1016/j.compag.2020.105542

Bao, W., Fan, T., Hu, G., Liang, D., and Li, H. (2022). Detection and identification of tea leaf diseases based on AX-RetinaNet. *Sci. Rep.* 12, 1–16. doi: 10.1038/s41598-022-06181-z

Barbedo, J. G. (2018). Factors influencing the use of deep learning for plant disease recognition. *Biosyst. Eng.* 172, 84–91. doi: 10.1016/j.biosystemseng.2018.05.013

Bergstra, J., and Bengio, Y. (2012). Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* 13, 281–305.

Bi, L., and Hu, G. (2020). Improving image-based plant disease classification with generative adversarial network under limited training set. *Front. Plant Sci.* 1945. doi: 10.3389/fpls.2020.583438

Brahimi, M., Arsenovic, M., Laraba, S., Sladojevic, S., Boukhalfa, K., and Moussaoui, A. (2018). "Deep learning for plant diseases: detection and saliency map visualisation," in *Human and machine learning* (Cham, Switzerland: Springer), 93–117.

Brahimi, M., Boukhalfa, K., and Moussaoui, A. (2017). Deep learning for tomato diseases: classification and symptoms visualization. *Appl. Artif. Intell.* 31, 299–315. doi: 10.1080/08839514.2017.1315516

Brahimi, M., Mahmoudi, S., Boukhalfa, K., and Moussaoui, A. (2019) "Deep interpretable architecture for plant diseases classification," in *2019 signal processing: Algorithms, architectures, arrangements, and applications (SPA)* (Poznan, Poland: IEEE), 111–116.

Chen, J., Chen, J., Zhang, D., Sun, Y., and Nanehkaran, Y. A. (2020a). Using deep transfer learning for image-based plant disease identification. *Comput. Electron. Agric.* 173, 105393. doi: 10.1016/j.compag.2020.105393

Chen, J., Zhang, D., and Nanehkaran, Y. A. (2020b). Identifying plant diseases using deep transfer learning and enhanced lightweight network. *Multimedia Tools Appl.* 79, 31497–31515. doi: 10.1007/s11042-020-09669-w

Chen, J., Zhang, D., Nanehkaran, Y. A., and Li, D. (2020c). Detection of rice plant diseases based on deep transfer learning. *J. Sci. Food Agric.* 100, 3246–3256. doi: 10.1002/jsfa.10365

Dai, J., Li, Y., He, K., and Sun, J. (2016). R-fcn: Object detection *via* region-based fully convolutional networks. *Adv. Neural Inf. Process. Syst.* 29, 379–387.

Fan, X., Luo, P., Mu, Y., Zhou, R., Tjahjadi, T., and Ren, Y. (2022). Leaf image based plant disease identification using transfer learning and feature fusion. *Comput. Electron. Agric.* 196, 106892. doi: 10.1016/j.compag.2022.106892

Gui, P., Dang, W., Zhu, F., and Zhao, Q. (2021). Towards automatic field plant disease recognition. *Comput. Electron. Agric.* 191, 106523. doi: 10.1016/j.compag.2021.106523

Haque, M., Marwaha, S., Deb, C. K., Nigam, S., Arora, A., Hooda, K. S., et al. (2022). Deep learning-based approach for identification of diseases of maize crop. *Sci. Rep.* 12, 1–14. doi: 10.1038/s41598-022-10140-z

Hassan, S. M., Maji, A. K., Jasiński, M., Leonowicz, Z., and Jasińska, E. (2021). Identification of plant-leaf diseases using CNN and transfer-learning approach. *Electronics* 10, 1388. doi: 10.3390/electronics10121388

He, H., and Ma, Y. (2013). *Imbalanced learning: foundations, algorithms, and applications*. (Hoboken, NJ, USA: Wiley-IEEE Press)

He, K., Zhang, X., Ren, S., and Sun, J.. (2015) "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, (Washington, USA: IEEE Computer Society. p. 1026–1034.

Hinton, G., Srivastava, N., and Swersky, K. (2012). Neural networks for machine learning. *Coursera video lectures* 264, p. 2146–2153.

Hughes, D., and Salathé, M. (2015). An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv preprint arXiv:1511.08060.* doi: 10.48550/arXiv.1511.08060

Ioffe, S., and Szegedy, C.. (2015) "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning* (Lille, France: PMLR), 448–456.

Jiang, Z., Dong, Z., Jiang, W., and Yang, Y. (2021). Recognition of rice leaf diseases and wheat leaf diseases based on multi-task deep transfer learning. *Comput. Electron. Agric.* 186, 106184. doi: 10.1016/j.compag.2021.106184

Kamal, K., Yin, Z., Wu, M., and Wu, Z. (2019). Depthwise separable convolution architectures for plant disease classification. *Comput. Electron. Agric.* 165, 104948. doi: 10.1016/j.compag.2019.104948

Kingma, D. P., and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980.* doi: 10.48550/arXiv.1412.6980

Krishnaswamy Rangarajan, A., and Purushothaman, R. (2020). Disease classification in eggplant using pre-trained VGG16 and MSVM. *Sci. Rep.* 10, 1–11. doi: 10.1038/s41598-020-59108-x

Lee, S. H., Goëau, H., Bonnet, P., and Joly, A. (2020). New perspectives on plant disease characterization based on deep learning. *Comput. Electron. Agric.* 170, 105220. doi: 10.1016/j.compag.2020.105220

Liang, W.-J., Zhang, H., Zhang, G.-F., and Cao, H.-X. (2019). Rice blast disease recognition using a deep convolutional neural network. *Sci. Rep.* 9, 1–10. doi: 10.1038/s41598-019-38966-0

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017). Focal loss for dense object detection. *Proceedings of the IEEE international conference on computer vision*. (Venice, Italy: IEEE Computer Society) p. 2980–2988.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., et al. (2016) "Ssd: Single shot multibox detector," in *European Conference on computer vision* (Amsterdam, Netherlands: Springer), 21–37.

Liu, J., and Wang, X. (2020). Tomato diseases and pests detection based on improved yolo V3 convolutional neural network. *Front. Plant Sci.* 11, 898. doi: 10.3389/fpls.2020.00898

Liu, J., and Wang, X. (2021). Plant diseases and pests detection based on deep learning: a review. *Plant Methods* 17, 1–18. doi: 10.1186/s13007-021-00722-9

Olsen, A., Konovalov, D. A., Philippa, B., Ridd, P., Wood, J. C., Johns, J., et al. (2019). DeepWeeds: A multiclass weed species image dataset for deep learning. *Sci. Rep.* 9, 1–12. doi: 10.1038/s41598-018-38343-3

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* 28, 91–99. doi: 10.48550/arXiv.1506.01497

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747.* doi: 10.48550/arXiv.1609.04747

Rukundo, O. (2021). Effects of image size on deep learning. *arXiv preprint arXiv:2101.11508.* doi: 10.48550/arXiv.2101.11508

Saleem, M. H., Khanchi, S., Potgieter, J., and Arif, K. M. (2020). Image-based plant disease identification by deep learning meta-architectures. *Plants* 9, 1451. doi: 10.3390/plants9111451

Saleem, M. H., Potgieter, J., and Arif, K. M. (2022a). A performance-optimized deep learning-based plant disease detection approach for horticultural crops of new Zealand. *IEEE Access* 10, 89798–89822. doi: 10.1109/ACCESS.2022.3201104

Saleem, M. H., Velayudhan, K. K., Potgieter, J., and Arif, K. M. (2022b) Weed identification by single-stage and two-stage neural networks: A study on the impact of image resizers and weights optimization algorithms. *Front. Plant Sci.* 920. doi: 10.3389/fpls.2022.850666

Tan, M., Pang, R., and Le, Q. V. (2020) "Efficientdet: Scalable and efficient object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*. (Seattle, WA, USA: IEEE) p. 10781–10790.

Thangaraj, R., Anandamurugan, S., and Kaliappan, V. K. (2021). Automated tomato leaf disease classification using transfer learning-based deep convolution neural network. *J. Plant Dis. Prot.* 128, 73–86. doi: 10.1007/s41348-020-00403-0

Too, E. C., Yujian, L., Njuki, S., and Yingchun, L. (2019). A comparative study of fine-tuning deep learning models for plant disease identification. *Comput. Electron. Agric.* 161, 272–279. doi: 10.1016/j.compag.2018.03.032

Wang, X., Liu, J., and Zhu, X. (2021b). Early real-time detection algorithm of tomato diseases and pests in the natural environment. *Plant Methods* 17, 1–17. doi: 10.1186/s13007-021-00745-2

Wang, P., Niu, T., Mao, Y., Zhang, Z., Liu, B., and He, D. (2021a). Identification of apple leaf diseases by improved deep convolutional neural networks with an attention mechanism. *Front. Plant Sci.* 1997. doi: 10.3389/fpls.2021.723294

Who, J., and Consultation, F. E. (2003). Diet, nutrition and the prevention of chronic diseases. *World Health Organ Tech Rep. Ser.* 916, 1–149.

Wu, X.-Z., and Zhou, Z.-H. (2017) "A unified view of multi-label performance measures," in *International conference on machine learning* (Sydney, Australia: PMLR), 3780–3788.

Xie, X., Ma, Y., Liu, B., He, J., Li, S., and Wang, H. (2020). A deep-learning-based real-time detector for grape leaf diseases using improved convolutional neural networks. *Front. Plant Sci.* 11, 751. doi: 10.3389/fpls.2020.00751

Chapter 9 Conclusions and Future Outlook

9.1. Conclusions

This Ph.D. research has produced 8 journal articles, 6 out of which have already been published in Q1 journals, and the remaining 2 are under review. The main objectives of this research were to propose deep learning (DL)-based approaches for the localization and classification of diseases and weeds in several plant species. In this regard, the performance improvement of the state-of-the-art DL models was successfully achieved. From the results and analysis presented in this thesis, DL has proven to be a flexible and robust technology, since a method/approach proposed for one plant-based agricultural operation like weed detection, was also utilized for other applications such as plant disease identification, and vice versa. Although each chapter provided in this thesis highlights the scientific significance, a summary of the research outcomes is provided as under:

- A comprehensive comparative analysis was presented for 18 deep learning models and the best-obtained models were optimized with DL optimization algorithms by using the PlantVillage dataset. The Xception model trained with the Adam optimizer achieved the highest validation accuracy and the F1-score of 99.81% and 0.9978, respectively, which were slightly higher than the default settings and better than the models proposed in the previous studies.
- Next, a comprehensive analysis of various DL meta-architectures was performed for both the classification and localization of plant diseases. This phase of the research also evaluated the significance of DL optimizers on the model's performance. The Single Shot MultiBox Detector (SSD) trained with Adam obtained the highest mean average precision (mAP) of 73.07%. It was 6.56% better than the one obtained with the default optimizer (SGD with momentum).
- A comprehensive DL-based weed detection pipeline was presented. Essential/early steps were taken before modifying state-of-the-art DL model to identify weeds under real agricultural conditions, using the DeepWeeds dataset. The Faster Region-based Convolutional Neural Network (R-CNN) ResNet-101 model attained the highest mAP with a final value of 93.44%, which was 5.8% better than the one obtained with the default settings.
- Using the outcomes obtained through the above research, an in-depth analysis of the best-obtained model named Faster R-CNN ResNet-101 was performed. A new enhanced anchor box

approach was presented. The resulting modification to the anchor box contained an addition of a 64x64 scale size; replaced the default aspect ratios with 1:3 and 3:1. These modifications improved the detection of the targeted Chinee Apple weed to 93.57% with an enhancement of 24.95% in average precision (AP) and 2.58% in mAP.

- The performance-optimized DL approach was presented for the identification of plant diseases using a novel NZDLPlantDisease-v1 dataset. The idea presented for weed detection was adopted for this research. Furthermore, an in-depth analysis of position-sensitive score maps and anchor boxes of the Region-based Fully Convolutional Network (R-FCN) model (the best-obtained DL model) was performed. The final mAP obtained by the optimized R-FCN model was 93.80%, which was 19.33% better than the default settings. Furthermore, the effectiveness and robustness of the research were demonstrated with the five-fold stratified cross-validation technique and tested on an externally generated dataset. This research remarkably showed an extended potential of a complex plant-based agricultural problem like plant disease detection, considering various practical agricultural problems, such as, detection of plant disease in multiple organs, presence of multiple disease in plant organs at a time, and significance of the DL-based optimized model for the identification of disease in various crops.
- The final study comprises a transfer learning approach through optimized weights obtained on three agricultural datasets. The Faster R-CNN Inception ResNet-v2 was found the best-suited DL model for a newly generated dataset called NZDLPlantDisease-v2. Later, the weights optimized through the PlantVillage dataset improved the results of the Faster R-CNN model for NZDLPlantDisease-v2. The final mAP was found to be 91.33% which was improved by 9.25%, compared to the default settings.

9.2. Outlook

The research outcomes of the presented methodologies have generated numerous opportunities for future research. The approaches/methods proposed in this thesis can be applied in other agricultural operations such as fruit harvesting/recognition, plant identification, classification of agricultural land cover, analysis of soil moisture content, and severity analysis of defected plants.

Future research could also comprise segmentation-based models to perform pixel-wise identification of plant diseases and weeds. Similarly, various advanced sensing technologies, such as microfluidics technology, can be integrated with the optimized/modified DL architectures using the proposed datasets. Moreover, an integration of various imaging techniques like hyperspectral/multispectral imaging with DL models could also be employed to get further advancements in both selected

agricultural operations. Furthermore, the recent advancement in deep learning through CapsuleNet can also be explored and compared with Convolution Neural Networks (CNNs).

The image-based analysis was presented throughout this thesis. However, real-time detection could also be performed using the proposed methodologies. Furthermore, the detection time could be attempted to reduce by proposing modifications in the DL model. In addition, a complete crop protection system could be designed consisting of continuous crop monitoring and intelligently guiding the farm workers to take appropriate and timely action using a webpage or mobile application.

Furthermore, robotic systems can be developed for use in horticultural fields utilizing datasets and optimized DL models. These systems would require a team effort of engineers/data scientists and agronomists to apply crop protection remedies such as the application of fungicide sprays on defected parts of plants, the removal of disease organs, and the application of herbicides on weeds. The robotic system could reduce the overall cost of crop protection. The latest research topic on the Internet of Robotic Things (IoRT) should also be explored for agricultural purposes to further advance the field of research. Moreover, a deep learning-based robot can be used to take a survey of the plants to analyze their nutrient deficiency and the maturity of the crops.

The artificial intelligence part for an agricultural robot could be taken from the optimized/modified DL models presented in this thesis. Further components of the robotic system could be proposed, such as a soft and flexible robotic hand gripper/manipulator. These grippers could be designed to detach defective parts of the plant and a robotic base could be built that can be operated in a real agricultural environment.

The proposed novel dataset contains diseases in multiple organs of several species of plants. However, some crops contain disease only in their leaves or stems. Therefore, these datasets should be extended to other plant organs and crops. Furthermore, the images can be collected in the agricultural environment of different countries to obtain further diversity in the symptoms of plant disease.

Appendix 1 Automation in Agriculture by Machine and Deep Learning Techniques: A Review of Recent Developments

This appendix contains content from the following article along with the published correction in the article.

M. H. Saleem, J. Potgieter, and K. M. Arif, "Automation in agriculture by machine and deep learning techniques: A review of recent developments," Precision Agriculture, vol. 22, no. 6, pp. 2053-2091, 2021.

Precision Agriculture (2021) 22:2053–2091
https://doi.org/10.1007/s11119-021-09806-x

REVIEW

Automation in Agriculture by Machine and Deep Learning Techniques: A Review of Recent Developments

Muhammad Hammad Saleem[1] · Johan Potgieter[2] · Khalid Mahmood Arif[1]

Accepted: 10 April 2021 / Published online: 21 April 2021

Abstract
Recently, agriculture has gained much attention regarding automation by artificial intelligence techniques and robotic systems. Particularly, with the advancements in machine learning (ML) concepts, significant improvements have been observed in agricultural tasks. The ability of automatic feature extraction creates an adaptive nature in deep learning (DL), specifically convolutional neural networks to achieve human-level accuracy in various agricultural applications, prominent among which are plant disease detection and classification, weed/crop discrimination, fruit counting, land cover classification, and crop/plant recognition. This review presents the performance of recent uses in agricultural robots by the implementation of ML and DL algorithms/architectures during the last decade. Performance plots are drawn to study the effectiveness of deep learning over traditional machine learning models for certain agricultural operations. The analysis of prominent studies highlighted that the DL-based models, like RCNN (Region-based Convolutional Neural Network), achieve a higher plant disease/pest detection rate (82.51%) than the well-known ML algorithms, including Multi-Layer Perceptron (64.9%) and K-nearest Neighbour (63.76%). The famous DL architecture named ResNet-18 attained more accurate Area Under the Curve (94.84%), and outperformed ML-based techniques, including Random Forest (RF) (70.16%) and Support Vector Machine (SVM) (60.6%), for crop/weed discrimination. Another DL model called FCN (Fully Convolutional Networks) recorded higher accuracy (83.9%) than SVM (67.6%) and RF (65.6%) algorithms for the classification of agricultural land covers. Finally, some important research gaps from the previous studies and innovative future directions are also noted to help propel automation in agriculture up to the next level.

Keywords Agricultural robotics · Machine learning · Deep learning · Convolutional neural network · Plant disease detection · Fruit harvesting

✉ Khalid Mahmood Arif
k.arif@massey.ac.nz

[1] Department of Mechanical and Electrical Engineering, School of Food and Advanced Technology, Massey University, Auckland 0632, New Zealand

[2] Massey Agritech Partnership Research Centre, School of Food and Advanced Technology, Massey University, Palmerston North 4442, New Zealand

Introduction

The agricultural industries are facing several problems including deficiency in the growth of products like fruits, vegetables, etc. (Chen et al., 2019), unpredictable soil contents (Padarian et al., 2019), improper application of pesticides (Sladojevic et al., 2016), herbicides, fungicides or insecticides to reduce crop/plant diseases and shortage of trained/skilled labour (Zhao et al., 2016a), etc. It is very important to address these issues as advancements in agriculture play a vital role in the economy of a country. Just like other fields of research including medical science, mechanical/automation, and business industries, etc., agriculture can also benefit from the use of robots to complement the human workforce. Therefore, in recent years, several attempts have been made to resolve agricultural issues through robotic platforms (Ebrahimi et al., 2017; Wspanialy & Moussa, 2016; Zhao et al., 2016a). Many state-of-the-art approaches have been introduced/modified to perform various agricultural tasks like fuzzy logic/classifier (Cho, Chang, et al., 2002; Cho, Lee, et al., 2002; Sujaritha et al., 2017), combined radar-vision system (Milella et al., 2011), HIS colour model (Feng et al., 2015), improved Otsu threshold algorithm (Wei et al., 2014), integration of various sensors (Milella, Reina, et al., 2019; Reina et al., 2016), self-supervised scheme (Reina et al., 2016), etc. In this regard, Artificial Intelligence (AI) has been proven to have great potential towards agricultural applications by the implementation of robotic systems with machine learning (ML)/deep learning (DL) algorithms (Ebrahimi et al., 2017; McCool et al., 2017; Zhang, Jia, et al., 2018; Zhang, Qiao, et al., 2018). Some advanced visualization techniques are prominent: saliency map visualization (Brahimi et al., 2018), hyperspectral imaging (Mahlein et al., 2017; Wang, Vinson, et al., 2019; Wang, Zhang, et al., 2019), multispectral imaging (Patrick et al., 2017; Pourazar et al., 2019; Slaughter et al., 2008) and thermal imaging (Azouz et al., 2015; Ishimwe et al., 2014), etc., have also been applied with ML/DL models for agricultural tasks. Therefore, with the progress in AI, the performance of many complex agricultural operations has improved as compared to the earlier approaches. This led us to present an overall review of research outcomes that have been obtained for agricultural applications by the implementation of ML/DL algorithms through robotic systems.

Some review articles have been published incorporating only a particular type of agricultural application with/without a robotic system by considering AI/computer vision/other advanced vision control techniques. For example, a recent review addressed the crop water stress by the machine learning approach (Virnodkar et al., 2020). A review article summarized the statistical ML algorithms, which have been implemented for various agricultural operations (Rehman et al., 2019). In (Huang et al., 2010), soft computing techniques including fuzzy logic, neural network, genetic algorithm, decision tree, and support vector machine (SVM) were presented for the analysis of soil, precision agriculture, and management of crops. A comprehensive review was conducted for precision agriculture by Unmanned Aerial Systems (UAS) and important future directions were also provided in the article (Zhang & Kovacs, 2012). In (Kamilaris & Prenafeta-Boldú, 2018), the DL architectures were reviewed for several agricultural operations. The review presented in (Zhao et al., 2016a) indicated the algorithms/schemes developed for vision control of harvesting robots. Another review paper outlined the harvesting robots to show their performance along with the procedures of robotic designs, and adaptive algorithms for harvesting purposes. Some interesting future recommendations including modification in the environment of crops, innovative robotic designs, and other important factors like safety and economy were also summarized (Bac et al., 2014). For the harvesting purpose, the advancement

in sensors was summarized in (Zujevs et al., 2015), by dividing them into four classes: chemical, tactile, proximity sensors, and computer vision. The issues like an in-camera sensor, design of the filter, and image segmentation methods for the identification of fruits through harvesting robots were presented in Li et al. (2011). Another review article presented the development of sensors for the detection/localization of fruit; it also described the AI-based classification methods and highlighted loopholes in those approaches (Gongal et al., 2015). The applications of machine vision with AI for agricultural tasks like detection of disease/pests in crops, evaluation of the quality of the grain, and automatic detection of plant phenotyping were studied in (Patrício & Rieder, 2018). The procedure for weed detection by various classification methods including machine learning and deep learning was reviewed in (Wang, Vinson, et al., 2019; Wang, Zhang, et al., 2019). The supervision of plant pathology by the robotic system while utilizing AI and machine vision techniques were presented in (Ampatzidis et al., 2017). Various sensing technologies and advanced cameras along with their limitations to categorize fruit/plant and analyse the physical structure of plants were summarized in (Narvaez et al., 2017). Another review article outlined the latest smart methodologies like internet of things (IoT), ML, and DL for agricultural purposes including crops/plant disease, pesticide and weed control, and storage and water management. (Jha et al., 2019). A review paper summarized ML algorithms for addressing weed detection, plant disease/pest detection tasks (Behmann et al., 2015). A recent review article presented the DL-based techniques for various agricultural applications (Santos et al., 2019). Another review paper explained and summarized deep learning models for the identification and classification of plant disease along with the application of DL with advanced imaging techniques including hyperspectral/multispectral imaging and some interesting future directions were also provided (Saleem et al., 2019). Moreover, the application of Big Data for agriculture was reviewed by Wolfert et al. (2017).

To the best of the authors' knowledge, there is no systematic review in a single article presenting the performance of robotic systems by machine/deep learning algorithms considering the major agricultural operations including detection of plant disease, identification of crop/plant, fruit counting, fruit recognition, identification of weed, crop/weed discrimination, and classification of agricultural land cover. Therefore, this review article will be useful to advance the agricultural field of research by studying machine and deep learning techniques that have been implemented on various intelligent agricultural systems. It will also be helpful to understand the research gaps in several complex agricultural applications to save cost related to agricultural protection and increase the growth of several agricultural products. To understand an overall idea of a robotic system for agricultural operations by implementing an ML/DL algorithm, Fig. 1 can be a good resource. First, the agricultural application should be selected, which would lead to the selection of a certain robotic platform that can be primarily used for the collection of datasets. Then, the Machine Learning/Deep Learning model would be proposed and trained into a robot that will perform the agricultural task, and finally record the accuracy of the models in terms of various performance metrics, like classification accuracy, F1-score, detection/failure rate, etc.

On top of that, during this review, the following questions were addressed that will guide the researchers of agricultural automation about many aspects of ML/DL algorithms employed through robotic platforms specifically in agricultural fields.

- Which agricultural operations have majorly implemented machine/deep learning algorithms through automated systems and what are the robotic platforms adopted for these agricultural tasks?

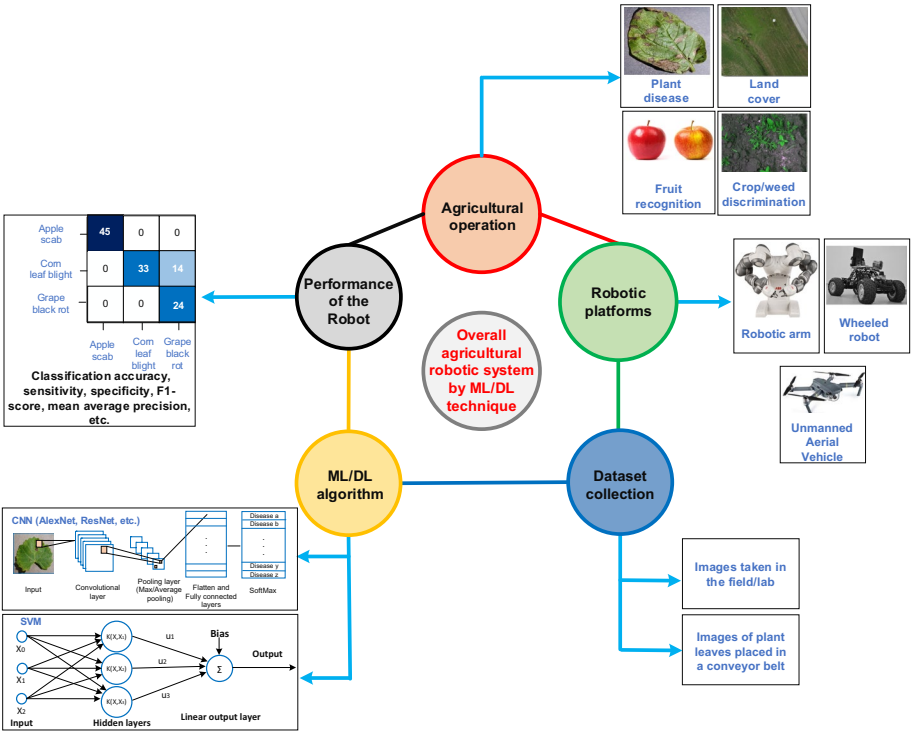


Fig. 1 Block diagram of the implementation of robotic system through ML/DL algorithms

- Which agricultural products/fruits/vegetables have been included in the previous studies when considering the implementation of ML and DL algorithms for robotic systems?
- Which ML/DL algorithms have been applied frequently for agricultural operations?
- How much has deep learning outperformed traditional machine learning algorithms for various agricultural tasks?
- Which performance metrics have been considered in the previous studies for the evaluation of ML and DL models that were used to perform agricultural tasks?
- What are the research gaps which could be filled to achieve better performance of various agricultural operations by ML/DL-based automated systems?

The remainder of the paper is further divided into the following sections: "Application of Traditional Machine Learning Algorithms in Agricultural Robots" presents machine learning models for various agricultural applications applied on robotic systems along with the research gaps; "Deep Learning Approach for Agricultural Operations by Robotic Platforms" elaborates the deep learning architectures for several agricultural operations implemented through robotic platforms along with the performance plots, and "Conclusion and Future Directions" concludes the review along with some future directions which will be helpful to achieve higher accuracy and great advancements in several agricultural tasks.

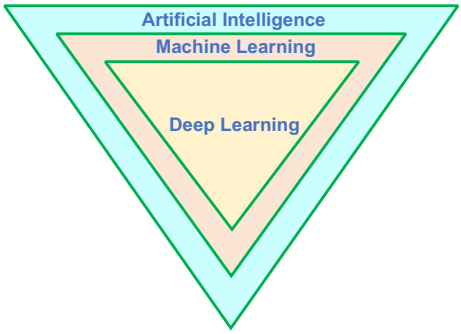


Fig. 2 A hierarchy of artificial intelligence (AI) according to which machine learning is typically a subset of AI and similarly deep learning is the subcategory of machine learning

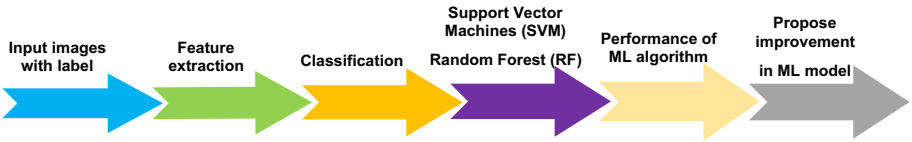


Fig. 3 A general representation of steps to implement machine learning algorithms

Application of Traditional Machine Learning Algorithms in Agricultural Robots

In this era of automation, Artificial Intelligence (AI) has complemented the human workforce in many real-life applications. Similarly, the agricultural industries also require smart solutions to address important issues like saving cost, better production of agricultural products like fruits/vegetables, shortage of trained labour, etc. In this regard, Machine Learning (ML) as a subset of AI produced a significant contribution to agricultural automation. The ML has further subcategories such as Deep Learning (DL) , whichis an emerging technology to perform various agricultural operations intelligently. Ageneral representation of AI, ML, and DL is presented in Fig. 2.

Before the evolution of deep learning architectures, prominent among which is AlexNet (Krizhevsky et al., 2012), machine learning (ML) algorithms produced many state-of-the-art results for various agricultural tasks. In these algorithms, Support Vector Machines (SVM), K-Nearest Neighbour (KNN), Random Forest (RF) classifier, and Decision Tree (DT) are the most prominent models. Although the ML algorithms like SVM have also been used to perform various complex tasks like classification and mapping of agricultural terrain (Reina et al., 2017), this review is focused on the studies which applied ML models/algorithms to perform five agricultural tasks through robotic systems as described in the next sub-sections. Also, their limitations are summarized which could help to advance upcoming future research in the field of agricultural automation. The general flow for the implementation of an ML algorithm is presented in Fig. 3.

Plant Disease/Pest Detection and Classification

The diseases/pests on plant species produce a significant impact on the growth of agricultural products. Therefore, their detection and classification are a necessity, particularly by an automated approach. In this regard, ML algorithms have been applied to perform this important agricultural task. For example, a Multi-Support Vector Machine (M-SVM) was proposed for the detection of disease in citrus fruit and its performancewas compared with state-of-the-art approaches like Weighted K-Nearest Neighbour (W-KNN), Decision Tree (DT), Linear Discriminant Analysis (LDA), and EnsembleBoosted Tree (EBT). However, the model were not comparedwith DL architectures toprove the effectiveness of the model more clearly (Sharif et al., 2018). Another researchwas conducted to detect and classify the healthy anddiseased leaves of vine by Local Binary Patterns and One-Class Classifiers (Pantaziet al., 2019).

The ML algorithms have also been implemented through robotic platforms for the detection of plant disease. For example, an Unmanned Aerial Vehicle (UAV) was used for the detection of Citrus greening and well-known ML algorithms like linear SVM, coarse gaussian SVM, standard gaussian SVM, K-Nearest Neighbour, and simple and complex Decision Tree were implemented to obtain the best-suitable model. From this study, a research gap can be filled by comparing the performance of ML models with well-known DL models like AlexNet, ResNet-50, VGG-16, etc. for the classification between healthy and diseased leaves (Sarkar et al., 2016). A mobile robot was implemented in a strawberry greenhouse to detect its disease; an SVM algorithm was applied for this purpose and achieved a considerably lower prediction error (Ebrahimi et al., 2017).

Plant/Leaves Recognition and Classification

Another important task of plant recognition has been done by state-of-the-art ML techniques through robotic platforms. A mobile robot was implemented to find its best route for a plantation in a real agricultural farm (Jodas et al., 2013) and for that purpose, SVM and ANN were evaluated and achieved 93% and 90% accuracy respectively. A critical task of the classification of grapevines was performed by SVM and ANN models through an all-terrain vehicle (Gutiérrez et al., 2018). This research could have a more interesting analysis if the performance of these two ML algorithms were compared with some successful CNN models like AlexNet, VGG, ResNet-50, etc. Another plant classification-related task was described in Huang et al. (2016) which proposed and designed a stand-still imaging system consisting of a hyperspectral camera, and a Least Squares Support Vector Machine (LSSVM) model was selected to classify the maize seed. Although the technique implemented in this research achieved good classification accuracy (CA), still the effectiveness of LSSVM should be proved by comparing its performance with the other ML classifiers like RF. Furthermore, more diversity in data samples should be included to prove the robustness of the applied model. In a research, as a prerequisite for an agricultural robot in a practical field, an SVM-based classification method was applied to distinguish eight different plant species (Dyrmann et al., 2018); the method proposed in this research improved the classification accuracy which showed the significance of the work. Another research performed the classification among six different plant species by a BoniRob mobile robot through the implementation of well-known ML algorithms. Their comparison brought simple logistic regression, SVM, and neural network, with the best results (Weiss et al., 2010).

A research used a UAV-based system for the task of tobacco plant recognition. An SVM model was implemented for this purpose and further enhancement in the performance was suggested by the candidate region extraction and feature extraction (Xie et al., 2016).

Crop/Weed Discrimination and Classification of Weeds/Crops

A critical agricultural task of crop and weed discrimination is also important to address as it is useful for determining the amount of herbicide required to control the weeds. Most of the studies were conducted for sugar beet fields; several were performed for carrot, rice, maize, and cereal farms. The ML algorithms including RF and SVM were prominently applied to robots for this agricultural operation. For example, the discrimination between crop and weed in a carrot farm was performed by an autonomous system through the implementation of the Random Forest classifier (Haug et al., 2014). For a more comprehensive assessment of the proposed system, there should be a comparative evaluation of this method with other state-of-the-art techniques like SVM. In (Cheng & Matson, 2015), an autonomous robot was used for the discrimination between crop (rice) and weed by featuring a base system consisting of a Harris Corner Detection algorithm along with ML algorithms that were compared in terms of precision and recall. A research was conducted for the discrimination between sugar beet and weeds by a BoniRob robotic platform; the classification was done by the Random Forest classifier, and the results were improved by MRF (Markov Random Field). Due to the successful classification outcomes, it is suggested that the RF would be useful for multiple-weed class problems (Lottes et al., 2016). Another research was conducted for the discrimination between a sugar beet crop and weed by a UAV; using information from RGB images; classification was done by RF classifier to achieve high precision and recall (Lottes et al., 2017). For an autonomous detection of weeds in a sugar beet field, two well-known algorithms—SVM and ANN—were implemented. The ANN achieved a considerably higher classification accuracy than SVM which proved the usefulness of the neural network-based technique (Bakhshipour & Jafari, 2018). A research was conducted to show the effectiveness of near-infrared mosaic hyperspectral imaging for crop and weed discrimination in a maize field. In the domain of machine learning, a random forest classifier was used (Gao et al., 2018); the higher precision and recall percentages showed that the applied method should be tested in a real-time robotic system. In (Tellaeche et al., 2011), the weeds were identified in a cereal crop by SVM and the classification accuracy was measured by Correct Classification Percentage (CCP) and Yule coefficient; the novelty of this work was shown by the evaluation of spray applied in the field. A research used the UAV for the detection/mapping of Silybum marianum weeds on hyperspectral images by comparing the performance of various ML techniques out of which One Class Support Vector Machine (OC-SVM) achieved the highest accuracy (Alexandridis et al., 2017).

Harvesting/Recognition of Fruits and Vegetables

The agricultural task of fruit harvesting has been addressed in recent studies that implemented well-known ML algorithms through robotic systems. However, the modified versions of ML models have also been proposed in a few research articles to perform this agricultural operation. For example, to identify tomatoes according to their maturity, a pixel and blob-based segmentation methods were applied along with a machine learning algorithm named X-means clustering which was derived from the famous K-means

clustering method (Yamamoto et al., 2014). In (Ji et al., 2012), a harvesting robot was tested in a real field environment by applying the vector median filter for the removal of noise, then image segmentation was applied for the extraction of the features of apples. An SVM-based method was applied to get improvement in recognition accuracy and some research gaps were also provided in the paper like addressing the unrecognized apples and reduction in timing of fruit recognition for a practical system. The apple harvesting system was developed in another research which consisted of a manipulator, end-effector, and vision system, whereas the SVM with RBF (radial basis function) was used for the recognition of apples, and the effectiveness of the system was shown by performing the experiments in the laboratory and real agricultural farms (De-An et al., 2011). To harvest tomatoes, an ML approach named RVM (Relevance Vector Machine) was introduced based on Bayesian inference, and a higher accuracy was obtained which has provided the motivation to use an RVM model for upcoming research (Wu, Zeng, et al., 2019; Wu, Zhang, et al., 2019). In addition to the applied methods, a comparative analysis should be provided in these studies with the other ML algorithms to show the usefulness of the proposed approach.

However, some research articles have considered various ML algorithms and compared their performance for the recognition/classification of fruits. For example, a conveyor belt-based system was proposed to evaluate various conditions of biscuits by implementing a Radial-based SVM classifier with Wilk's λ method which achieved a higher classification accuracy as compared to the Polynomial SVM and discriminant analysis (DA) (Nashat et al., 2011). In, (Tao & Zhou, 2017), the authors used the Colour-FPFH 3D descriptor to extract the features of apples. For the classification purpose, the Genetic Algorithm SVM classifier (GA-SVM) was used and its performance was compared with other classifiers like SVM, KNN, and RF. A complete study of Broccoli was presented in Kusumam et al. (2017), which incorporated the important steps from detection to size estimation and level of growth by a robotic tractor system and the SVM algorithm was again used along with a viewpoint feature histogram and temporal filter; a comparison was also done between KNN and SVM algorithms and the detection accuracy can be further improved by considering texture features. Another research used a tractor system for the localization and detection of Broccoli by using a method composed of VFH (ViewPoint Feature Histogram) and SVM (Support Vector Machine), its performance was increased when temporal filtering (TF) was included, and the proposed method was compared with ANN (Kusumam et al., 2016). Another study used the SVM classifier (Liu, Mao, et al., 2019; Liu, Pi, et al., 2019), which applied the HOG descriptor for the training of SVM and False Colour Removal (FCR) and Non-Maximum Suppression (NMS) were proposed for the removal of false positives and merge the overlapped detections. This research has practical importance for the future robotic system as the images were taken at 500–1000 mm distance which is quite feasible for an actual robotic platform. Therefore, the proposed method can be used in a real-time robotic system. Moreover, the proposed method was compared with the other approaches like AdaBoost (Zhao et al., 2016b), YOLO model (Redmon et al., 2016), Circular Gabor Filter, and Eigen Fruit (Kurtulmus et al., 2011).

Another approach used SVM for texture classification along with Canny edge detection with a graph-based connected component algorithm and the Hough line detection method for the removal/reduction of false positives of green citrus fruit (Sengupta & Lee, 2014). An SVM-based approach was proposed in (Mao et al., 2020) to recognize cucumbers in a farm; the method consisted of Iterative-RELIEF which was used for the extraction of colour components, background pre-processing being done by Median filter, Otsu algorithm, and Maximally Stable Extremal Regions (MSER); a fine-tuned DL model was proposed

for feature extraction and finally PCA was used for the reduction of the dimension which eventually became useful for SVM classification.

Land Cover Classification

Several researchers used well-known ML algorithms and compared their performance for the selection of the best-suited model to classify different classes of agricultural land covers. For example, the classification among agricultural lands was performed and compared by implementing DT, RF, and SVM and it was concluded that object-based SVM got the highest Overall accuracy (Duro et al., 2012). A research was conducted to classify 16 classes divided into ten agricultural and six non-agricultural landscapes; a comprehensive comparison was provided between six state-of-the-art ML techniques including Multilayer Perceptron (MLP), Support Vector Regression (SVR), the Least-Squares (LS)-SVM, Bagged Regression Trees (BaRTs), Boosted Regression Trees (BoRTs), and the Random Forest (RF) by using EPR- (Eenmalige perceels registratie—in the Dutch language) based data and CORINE Land-Cover 2006 dataset. It was found that SVM classifiers (SVR and LS-SVM) outperformed other classifiers in terms of pixel-level Nash–Sutcliffe (NS) index and some future directions were provided in the article including the selection of input variables and the implication of fractional abundance constraints (Heremans & Van Orshoven, 2015). For the land cover classification, three state-of-the-art methods were applied including Support Vector Machine (SVM), Neural Network (NN), and Classification and Regression Trees (CART). It was found that the SVM classifier achieved the highest classification accuracy (Shao & Lunetta, 2012). Another research performed a comparative study between RF, kNN, and SVM to classify six different classes (including agricultural landscape) by using images taken through Sentinel-2 satellite (Thanh Noi & Kappas, 2018). A study was conducted for the classification of croplands and this time TerraSAR-X satellite data was used; the significance of RF was noted by comparing its performance with Classification and Regression Tree (CART) (Sonobe et al., 2014). In (Peña et al., 2014), nine important crops were classified by considering input from images of ASTER satellite and state-of-the-art ML approaches like DT, LR, SVM, and MLP were utilized for this purpose. Among all of them, SVM and MLP outperformed the others and the authors implemented an SVM + SVM algorithm that achieved slightly higher accuracy than SVM and MLP models.

On the other hand, a few articles implemented only RF classifier for the classification of the landscape. For example, the Landsat-5 Thematic Mapper data was used to classify complex landscapes by RF algorithm and achieved 92% overall accuracy (Rodriguez-Galiano et al., 2012). Similarly, the research presented in Eisavi et al. (2015) showed the significance of the RF classifier by taking the images of 13 agricultural landscapes via Landsat 8 satellite.

The Random Forest (RF) classifier and Maximum Likelihood Classification (MLC) were implemented on images taken from SPOT 5 satellite for the classification of various agricultural cropland fields. The outcome of this research favoured RF classifier by a significant margin (Ok et al., 2012). A research performed the classification of four croplands by classical ML algorithms such as SVM and RF through images taken by time series UAV. The novelty of this work was proved by considering the effect of textural features through the Grey-Level Co-occurrence Matrix (GLCM) along with the spectral features. Moreover, DL architectures could also be applied for further improvement in the classification task (Kwak & Park, 2019).

Overall Presentation of ML Algorithms for Agricultural Operations by Robots

Few important research gaps/future directions related to each agricultural operation from this section are presented in Table 1. Moreover, a summary of the performance of ML algorithms is shown in Table 2.

Deep Learning Approach for Agricultural Operations by Robotic Platforms

After the development of deep learning (DL), many state-of-the-art models were implemented for various real-life applications. Among those models, Convolutional Neural Network (CNN) produced significant improvement for many image recognition/classification tasks. Similarly, agricultural operations have also been performed by the implementation of CNN architectures through robots.

Previously, some review articles were focused on DL with respect to certain agricultural operations. For example, a comprehensive review of DL in agriculture was presented in (Kamilaris & Prenafeta-Boldú, 2018), in which all the major agricultural tasks were summarized. In contrast, this review article presents deep learning approach for major agricultural operations implemented through robotic platforms. Moreover, few research articles are also included in this review which showed the effectiveness of proposed DL-based models for upcoming agricultural robotic projects. Furthermore, some important research gaps are mentioned to address agricultural issues by automation through CNN architectures. The performance plots are also drawn to indicate the significance of DL architectures over traditional/well-known ML models for the respective agricultural tasks.

The implementation of DL to perform agricultural operations through robots involves few steps as presented in FigS. 4, and 5 further explains all the three steps of Fig. 4 more clearly.

Plant Disease/Pest Detection and Classification

In recent times, DL has been considered a better method to perform agricultural tasks. These tasks are performed by implementing well-known CNN architectures or by proposing some modifications to those well-known models. A complex task of plant disease identification has been addressed by the DL techniques (Esgario et al., 2020; Li et al., 2020; Liu, Abd-Elrahman, et al., 2018; Liu, Zhang, et al., 2018; Singh et al., 2019). An overall review can be referred to (Saleem et al., 2019) related to plant disease identification by DL. However, in this section, a summary of the DL approaches applied through automated systems (like mobile robot, robotic arm, etc.) is provided for plant disease and pest identification.

An imaging system was proposed to detect the powdery mildew disease by the implementation of a famous CNN architecture named GoogLeNet and the accuracy was compared with experts' performance (Bierman et al., 2019). A research was conducted for the comparative evaluation of the performance of DL and ML algorithms for the detection of pests on tomato and pepper crops for autonomous robots and concluded with the superior accuracy of DL architecture, principally Faster RCNN (more accurate but requires more computation time) and SSD (less accurate than RCNN and requires less training time).

Table 1 Research gaps in some of the articles implemented ML algorithms to perform the respective agricultural tasks

Agricultural applications	Research gaps/future directions	Refs.
Plant disease detection	The proposed ML methods should be compared with the DL models to show the effectiveness of the applied algorithms more clearly	(Sarkar et al., 2016; Sharif et al., 2018)
Plant recognition	The applied ML algorithms should be tested in a real-time system	(Jodas et al., 2013)
	Other ML algorithms should also be considered like RF classifier. Furthermore, more diversity in data samples should be included to prove the robustness of the applied model	(Huang et al., 2016)
	The feature extraction should be improved in the future to further enhance the performance	(Xie et al., 2016)
Crop/weed discrimination	An efficient ANN model could further improve the performance as applied for other applications	(Tellaeche et al., 2011)
	The practical robotic system should be implemented and compared its performance with manual weeding system	(Haug et al., 2014)
	The robustness of RF can be shown by considering other crops	(Cheng & Matson, 2015)
	The ability of the RF classifier should be studied for multiple classes of weed	(Lottes et al., 2016)
Fruit/vegetables recognition and harvesting	Along with false positives, false negatives should also be considered in the future	(Yamamoto et al., 2014)
	The application of an advanced DL model could improve the recognition rate The recognition time should be reduced to make the practical system more feasible	(Ji et al., 2012)
	The features related to texture could be useful to get further improvements in detection	(Kusumam et al., 2017)
Land cover classification	The DL architectures could also be applied for further improvement in the classification task as these algorithms are very useful for image identification/classification tasks	(Kwak & Park, 2019)
	Some future directions were provided in the article including the selection of input variable selection and implication of fractional abundance constraints	(Heremans & Van Orshoven, 2015)
	The significance of RF can be proven more effectively by considering other ML algorithms like SVM	(Eisavi et al., 2015)

Appendix 1

Table 2 A summary of machine learning approaches used for various agricultural tasks by robotic systems/platforms along with their performance indicators and agricultural products

Agricultural applications	Agricultural products	Robotic platforms/systems	ML algorithms	Performance metrics with its % value		Refs.
Plant disease detection	Citrus	UAV	Linear SVM	Validation accuracy	93.3	(Sarkar et al., 2016)
			Standard Gaussian SVM		93.3	
			Complex DT		91.7	
			Simple DT		90.8	
			KNN		90	
			Coarse Gaussian SVM		88.3	
	Strawberry	Mobile robot	SVM	Mean percent error	2.25	(Ebrahimi et al., 2017)
Plant/leaves recognition and classification	Peanut	Mobile robot	SVM	Accuracy	93	(Jodas et al., 2013)
	Soybean		ANN		90	
	Grapevine	All-terrain vehicle	MLP	F1-score	99.05	(Gutiérrez et al., 2018)
			SVM		91.42	
	Maize	Stand still imaging system	LSSVM	Classification accuracy	94.4	(Huang et al., 2016)
	6 plant species	BoniRob mobile robot	LMT	Classification accuracy	98.8	(Weiss et al., 2010)
			Simple logistic function		98.8	
			IB1 (Lazy classifier)		92.04	
			BayesNet		89.62	
			Nnge		89.53	
			Hyperpipes		87.84	
	Tobacco	UAV	SVM	Accuracy	96.1	(Xie et al., 2016)

Table 2 (continued)

Agricultural applications	Agricultural products	Robotic platforms/systems	ML algorithms	Performance metrics with its % value		Refs.
Crop/weed discrimination	Carrot	BoniRob mobile robot	Random Forest	Classification accuracy	93.8	(Haug et al., 2014)
	Rice	BoniRob mobile robot	Decision Tree	Precision	98.2	(Cheng & Matson, 2015)
			SVM		95.3	
			Naïve Bayes		93.1	
	Sugar beet	BoniRob v3 mobile robot	Random Forest with Markov Random field	Precision	95	(Lottes et al., 2016)
	Sugar beet	UAV	Random Forest	Overall classification accuracy	96	(Lottes et al., 2017)
	Silybum marianum	UAV	One Class Support Vector Machine	Overall classification accuracy	96.05	(Alexandridis et al., 2017)
			One Class Self-Organizing Maps		94.65	
			Autoencoders		94.3	
			One Class Principal Component Analysis		90	
Fruits/vegetables harvesting/recognition	Apples	Harvesting mobile robot	SVM with colour and shape feature	Recognition success rate	89	(W. Ji et al., 2012)
	Apples	Mobile robot/manipulator	SVM with Radial Basis Function	Success rate	77	(De-An et al., 2011)
	Biscuits	Conveyor belt	SVM-R	Average classification accuracy	96.5	(Nashat et al., 2011)
			SVM-P		95.5	
			DA		94	
	Broccoli	Tractor	SVM (Viewpoint Feature Histogram (VFH) with Temporal Filtering)	Average precision	95.2	(Kusumam et al., 2017)
			SVM (VFH)		94.7	
			KNN (VFH)		93	

Table 2 (continued)

Agricultural applications	Agricultural products	Robotic platforms/systems	ML algorithms		Performance metrics with its % value			Refs.
Land cover classification	Crop, grassland	SPOT-5 HRG satellite	SVM		Overall accuracy	94.21		(Duro et al., 2012)
	Rock/soil, wetland,		Random Forest			93.39		
	Riparian and water		Decision Tree			88.84		
	10 agricultural and 6 non-agricultural land covers	Terra-MODIS	MLP	BaRT	Pixel-level NS index	45.6	45.3	(Heremans & Van Orshoven, 2015)
			SVR	RF		47.6	47.1	
			LS-SVM	BoRT		49.0	45.6	
	Almond	ASTER satellite	SVM + SVM	MLP + LR	Correct classification rate	91	88	(Peña et al., 2014)
	Sunflower							
	Walnut		SVM	DT + SVM		91	88	
	Tomato							
	Vineyard		SVM + MLP	LR + LR		90	87	
	Alfalfa		LR	DT + MLP		89	87	
	Corn		MLP + MLP	DT + LR		89	86	
	Rice		SVM + LR	DT		89	82	
	Safflower		MLP			88		

Table 2 (continued)

Agricultural applications	Agricultural products	Robotic platforms/systems	ML algorithms	Performance metrics with its % value		Refs.
	Tobacco, upland conifer forest, Corn,	Landsat Thematic Mapper-5	RF	Overall accuracy	92	(Rodriguez-Galiano et al., 2012)
	shrub-grasslands, Tropical, Greenhouse		CT		86	
	Olive, oak grove				86	
	Apple, vineyards,	Landsat 8’	Random Forest	Overall accuracy	91.82	(Eisavi et al., 2015)
	Grasslands, Urban, water, Bare land,					
	Wetland vegetation, corn, Summer crops,					
	Wheat, Salt area,					
	Road, Fallow,					
	Wheat, sugar beet, pepper	SPOT 5 satellite	Random Forest	Overall accuracy	85.89	(Ok et al., 2012)
	Rice, Corn, Tomato,		Maximum Likelihood		77.96	
	Highland Kimchi Cabbage, Potato	UAV	SVM	Overall accuracy	98.72	(Kwak & Park, 2019)
	Fallow, Cabbage,					

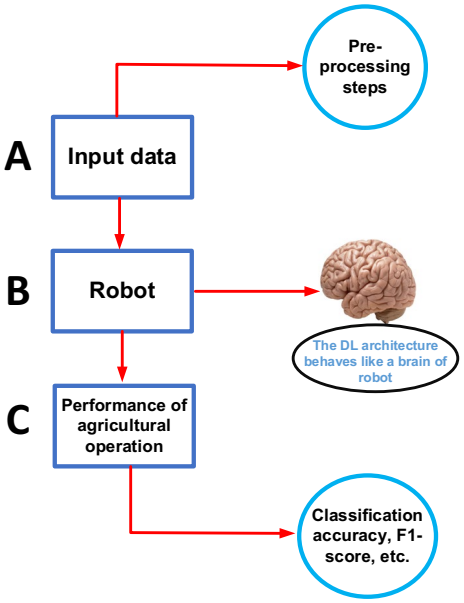


Fig. 4 The basic steps of a robotic platform for an agricultural task by DL approach; (**A**) indicates the input dataset like images of various plants/fruits, (**B**) presents that the robotic system needs a brain to perform certain tasks and here DL models act as a brain to the robot, and, finally, (**C**) represents the output of agricultural robots to show the significance of applied DL architecture

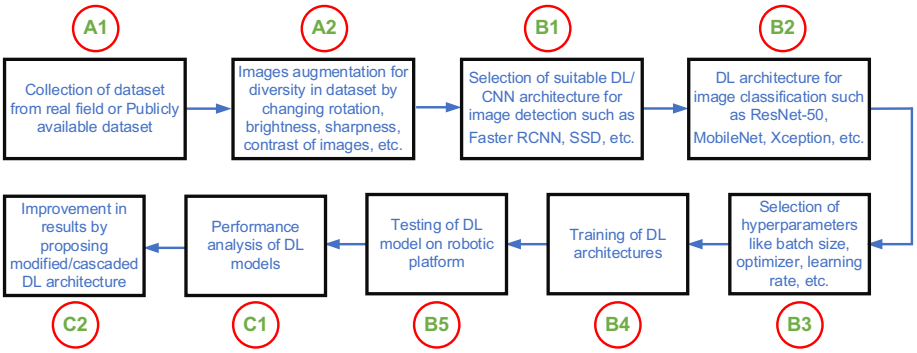


Fig. 5 A clearer explanation of (**A**, **B**, and **C**) (presented in the previous figure) require to implement an agricultural task by DL-based robot

Moreover, the characteristics of the automatic feature extraction of DL were presented that helped to achieve better accuracy/classification as compared to an ML approach which required complex feature engineering works, and some innovative future directions were also presented like data augmentation technique and an inclusion of dataset images having pests present in the plants to generate diversity in a dataset (Gutierrez et al., 2019). Another research was conducted to prove the significance of deep learning in terms of its ability to automatically detect disease on fruits through an automatic sorting machine; the performance of the proposed system could be improved by deep autoencoder (da Costa et al., 2020). For the detection of a crop virus, a Fully Convolution Neural Network (FCN) model was deployed on the hyperspectral images through a tractor-shaped system containing a push broom (Polder et al., 2019). A 6 degree-of-freedom (DoF) robotic arm was used to automatically detect the diseased leaves by the implementation of Faster-RCNN with

ResNet-101 model (Joffe et al., 2018). An Unmanned Aerial Vehicle (UAV)—based system was developed for the identification of vineyard disease by a deep learning algorithm applied to multispectral images, but it is suggested that false detections could be reduced by applying and testing various DL models (Kerkech et al., 2019). The UAV was also used for the detection of Fusarium wilt in a radish farm; the well-known VGG model performed well and achieved comparatively higher accuracy than the K-means clustering ML algorithm. Some recommendations were provided to improve the performance such as the combination of RGB and infrared images; a comprehensive analysis was recommended based on the severity of the disease (Ha et al., 2017). Another approach used UAV technology for the detection of disease in a radish farm by the implementation of K-means clustering along with GoogLeNet architecture (trained by a fine-tuning technique). The performance of the DL model was significantly better than the SVM model (Dang et al., 2018).

From this section, it can be concluded that various DL architectures were implemented for the detection of disease in plant leaves. However, the future research should be conducted to detect and classify the disease present in all the defected parts of the plant species including leaves, stem, fruit, and flowers, by utilizing the adaptive nature of DL. This is one of the most important research gaps provided in this review. Moreover, very few studies have been conducted to perform this task by a real-time automated system, therefore there is a need of a robotic platform than can address this agricultural problem. Furthermore, the chemical sprays like fungicide/herbicide/pesticides should be applied intelligently after the successful detection of plant disease, which would be helpful to generate a cost-effective crop protection system.

Plant/Leaves Recognition and Classification

Just like traditional ML algorithms, deep learning models have also applied for the plant recognition task. For instance, a research was conducted for the classification of several plant species by proposing/implementing CNN models (Dyrmann et al., 2016). A very important study was done in (Lee et al., 2017) to understand the concept and capability of deep learning models to extract the characteristics/features of several plant species. The development of deep plant phenomics (DPP) created a major contribution to the community of plant phenotyping (Ubbens & Stavness, 2017). Another state-of-the-art approach was proposed for the classification of plants with their multiple organs through CNN and RNN models (Lee et al., 2018). The classification of four different plant species was done by proposing a CNN model which outperformed the approaches like Scale-Invariant Feature Transform (SIRF) and speeded up robust features (SURF) (Kazerouni et al., 2019). A study was conducted for the classification of plant seedlings by CNN. This research also compared its performance with well-known ML techniques like SVM and KNN which proved the significance of the approach (Nkemelu et al., 2018).

Few of the studies were conducted to perform the task of plant recognition through robotic systems/platforms. To extract the stalk count and stalk width of the plant, a deep convolutional neural network and a semantic segmentation-based ground mobile robotic platform were proposed and validated the performance of the robot with two humans which showed the effectiveness of the proposed idea. The Faster Recurrent Convolution Neural Network (Faster RCNN) model was used for generating the bounding box and the binary output was obtained by Fully Convolution Network (FCN) to classify images as either stalk or background. As compared to human performance, the robot performed the stalk count task 30 times faster and stalk width measurement task 270 times faster (Baweja

et al., 2018). A study was conducted for the plant phenotyping by the deep learning technique based on a recently developed Point Cloud Network (PCN) model through multi-robotic systems (Wu, Zeng, et al., 2019; Wu, Zhang, et al., 2019). A research used the UAV to count corn plants through a DL model named U-Net. The successful result of this approach provided a future motivation to implement this type of system for other crops (Kitano et al., 2019). A robotic manipulator was used to recognize seven vegetables by famous DL feature extraction/detection architectures, but the recognition accuracy should be further increased by proposing some key modifications to the models applied in this research (Zheng et al., 2018). To recognize Legacy Blueberries plants, a Computer Numerical Control (CNC)—based system was developed and a CNN was proposed to achieve a good performance in terms of precision, recall, F1-score, and accuracy. Although this proposed scheme can also be tested for pest detection, more improvement in the system's performance is recommended in the future by the implementation of generative adversarial networks (GANs) for the generation of synthetic images (Quiroz & Alférez, 2020). The segmentation of Fig plants was done by UAV, and a CNN model was inspired by SegNet encoder-decoder architecture CNN. The code of the CNN model and dataset were published online for the research community. A good thing was that the complex and variable/original background of images were considered and it was suggested that the orthomosaic images could improve the system proposed in the paper (Fuentes-Pacheco et al., 2019). Another research used a UAV for the collection of datasets to detect Tobacco plants by the CNN models. Although the proposed CNN architecture achieved good accuracy, there is still room to improve the performance further by the implementation of various available well-known CNN models, the advanced training techniques like transfer learning/fine-tuning techniques could be utilized and some other crops should also be considered in the future (Fan et al., 2018). A research addressed the problem that occurs due to the critical distribution of heads of sorghum by a CNN model named RetinaNet on the UAV images; the system can achieve better performance by including diversity in the dataset (Ghosal et al., 2019). A hybrid approach consisting of SLIC (Simple Linear Iterative Clustering) and Hue properties was combined with a CNN model for the detection of flowers in a Soybean field and a single axis robot was used for this purpose, and the authors also provided a future direction—that seed pod counting should also be considered (Yahata et al., 2017). A research utilized the UAV for the detection, classification, counting of trees, and evaluation of varieties of citrus by the implementation of a famous DL detection model named YOLO-v3 (Ampatzidis & Partel, 2019).

In summary, the plant recognition task by DL models achieved considerably good performance. Some future works are recommended like diversity in datasets and considering different crops to prove the effectiveness of CNNs. And stalk count/width should also be addressed in more detail.

Crop/Weed Discrimination and Classification of Weeds/Crops

Another complex agricultural task of discrimination between crop and weed has been reported by the DL approach through real-time robotic systems. In (Adhikari et al., 2019), the authors presented a deep convolutional encoder-decoder neural network and achieved a higher mIoU (mean Intersection of Union) which was significantly higher than the previously-used models like UNet (Ronneberger et al., 2015), FCN (Long et al., 2015) and DeepLabV3 (Chen et al., 2017). A mobile platform was designed and implemented for the detection of weed in a radish farm by implementing an ANN model that showed the

obvious effectiveness of a neural network for the detection of weeds (SeI Cho, Chang, et al., 2002; Cho, Lee, et al., 2002). A CNN-based semantic segmentation for real-time crop and weed classification was done in a sugar beet field (Milioto et al., 2018). Another research was conducted to classify crop and weed by the implementation of a lightweight & deeper CNN on a mobile robot and the novelty of the work was that these CNN architectures were applied on the RGB along with Infra-red images (Potena et al., 2016). A research was conducted to propose a class-wise stem and pixel-wise semantic segmentation-based system for the stem and crop/weed classification. This research achieved state-of-the-art results through a mobile robot and UAV, and they outperformed conventional approaches like Random Forest, baseline-stem (Lottes et al., 2018a). An FCN model having an encoder-decoder structure was proposed and implemented in sugar beet fields through a mobile robot named BoniRob containing RGB and NIR cameras for the collection of datasets (Lottes et al., 2018b). An automated ground robotic system was implemented for crop and weed discrimination through a simple ANN model by considering the natural environment, ignoring plants having incomplete features, and maximizing the pixels of weeds (Jeon et al., 2011). The classification between crop and weed was also performed by the UAV platform through the famous CNN model ResNet and two agricultural fields were considered to show the effectiveness of the proposed system; supervised labelling was done to improve the AUC on both fields. Moreover, an improvement in background segmentation by using multispectral images and graphical interface to generate an infestation map was suggested to reduce costs while applying herbicide in the fields (Bah et al., 2018). A smart sprayer system was designed for the management of weeds and the system's performance was analysed by using two different Graphical Processing Units (GPUs); real/artificial plants were also considered which clearly proved the usefulness of the DL model for the detection of weeds. As a future direction, an algorithm should be deployed that can vary the amount of chemical spray required to control the weed and its performance should be compared with the traditional sprayers (Partel et al., 2019). A wheeled robot named AgBotII was implemented on a cotton field to manage the weeds and proposed an image locking system for a clustering algorithm. The value of the work was shown by introducing a new performance metric named DScore and discrimination of weed was done successfully without previous knowledge of the field (Hall et al., 2017). A Micro Aerial Vehicle (MAV) was also implemented for the treatment of weed in a sugar beet field. The images were taken by multispectral imaging technique and a recently-developed SegNet model was trained and tested for the classification of weed and crop which could achieve higher accuracy by training the model on a larger dataset (Sa et al., 2017). In (dos Santos Ferreira et al., 2017), a quadcopter was used for the collection of crop and weed images, whereas the classification task was done by a very famous CNN architecture called AlexNet on Caffe software. Classification accuracy was compared with ML-based state-of-the-art approaches like SVM, Random Forest, and AdaBoost, and the results were obtained under a controlled environment which leads to a research gap that can be filled by considering a real environment with a larger dataset. A research was conducted to generate a publicly available dataset for the classification of eight types of weeds that were trained and tested through well-known CNN models like Inception-v3 and ResNet-50 (Olsen et al., 2019). A mobile robot was used to generate the dataset for crop/weed detection (Di Cicco et al., 2017) and higher accuracy could be obtained by the use of NIR spectroscopy and hyperspectral imaging. A research was evaluated the robustness of the two models (JULE and DeepCluster) on the datasets developed in Olsen et al. (2019) and (dos Santos Ferreira et al., 2017) by unsupervised clustering algorithms (dos Santos Ferreira et al., 2019). A mobile robot was designed and implemented to classify crop and weed, and implemented

popular CNN architectures like AlexNet, VGG, ResNet, and Inception-v3, while the training was performed by the transfer learning technique through ImageNet dataset (Suh et al., 2018). It is recommended to use Multiple classes to better prove the strength of the applied method. In (Dyrmann et al., 2017), the DetectNet model proposed in Barker et al. (2016) was used for the detection of weeds in a wheat field and an all-terrain vehicle was operated to generate the dataset. The proposed DL model should be tested in a real-time system to show the effectiveness of DetectNet architecture. Another article was published in the domain of weed detection by comparing the performance between SVM and ResNet models through a UAV in sugar beet fields; the obtained results favoured the ResNet model (Bah et al., 2019).

To conclude, many UAVs and ground robots have been implemented on fields like sugar beet, corn, etc., for performing complex tasks of weed/crop discrimination by state-of-the-art DL models. These successful DL models should be tested on other crops by UAVs or other robotic platforms. Moreover, smart chemical sprayers should be deployed to control the weeds in agricultural fields.

Harvesting/Recognition of Fruits and Vegetables

Some of the recent studies were conducted for the fruit detection/harvesting task by well-known DL architectures or by proposing an improved version of a DL model for forthcoming agricultural robotic projects (Sa et al., 2016; Zhang et al., 2019). A few of them focused on designing a gripper for a fruit harvesting robot, like (Zhang, Harrison, et al., 2020; Zhang, Huang, et al., 2020) proposed a harvesting system consisting of a low-cost robotic gripper and manipulator; the detection of the fruits/vegetables was done by the state-of-the-art Mask-RCNN model. A research proposed an improved version of Faster-RCNN to detect fruits and the effectiveness of the proposed model was proved by comparing its performance with other well-known and successful DL image detection architectures including YOLO, Fast RCNN, and Faster RCNN (Wan & Goudos, 2020). A recent article proposed an apple recognition system by pulse couple neural network and genetic Elman neural network and achieved a higher recognition rate (Jia et al., 2020). In (Liu, Mao, et al., 2019; Liu, Pi, et al., 2019), the authors proposed an improved version of a DenseNet model (Huang et al., 2017) to recognize and harvest tomatoes in a real environment. That research used a complex/actual environment that proved the novelty of the research as many of the previous studies used plain/controlled background and the rate of detection was comparatively better when compared to popular CNN models like ResNet, DenseNet, and SSD architectures. Another recent research was conducted for the classification of date fruits by well-known AlexNet and VGG-16 models trained through the transfer learning technique and comparing the performance of these models with previously published work (Altaheri et al., 2019). For the tomato harvesting robot, a wavelet transform-based image processing technique was applied along with two hidden layer feed-forward neural network models (Arefi & Motlagh, 2013). Another research implemented DL architecture by proposing a CNN model to harvest tomatoes and obtained 91.9% accuracy in a short period of time (Zhang, Jia, et al., 2018; Zhang, Qiao, et al., 2018).

In this article, those studies which used different robotic platforms for fruit harvesting/recognition purpose are extensively summarized. A novel research was conducted in which a robotic manipulator consisting of four arms was designed and implemented in a kiwifruit orchard for harvesting; the novel end-effector for each arm was designed to pick kiwifruit safely and dynamic scheduling was also done. The detection of kiwifruits was

done by proposing a fully convolutional network named FCN-8S and real-field testing was performed which gave 51% successful harvesting results. Moreover, it was also determined that with the applied approach, the rate of successful harvesting can be increased to 70% and that a greater degree of freedom could increase cycle time (Williams et al., 2019). Few recent articles considered an important agricultural task of segmentation of fruit clusters in the real agricultural environment. In this regard, the first attempt was made for the estimation of canopy volume, counting, and detection of grape clusters. The images were taken by an RGBD camera placed on a mobile platform. Four pre-trained DL models were implemented; the VGG-19 model attained the highest accuracy (Milella, Marani, et al., 2019). Another recent study reported the segmentation of clusters of grapes by pre-trained DL architectures. Moreover, a novel method to improve the segmentation of cluster pixels was proposed. Due to high segmentation accuracy (Marani et al., 2020), this research could be adopted for the future research. A study implemented a simple backpropagation neural network on a sorting system consisting of a conveyor belt to classify the date fruit, and the future work should comprise an impact sensor, and feature a distribution-based method which should be introduced for better grading of the fruit (Al Ohali, 2011). A six DoF robotic manipulator was implemented for an ice lettuce farm and achieved a good success rate for harvesting, but the average cycle was comparatively slower than a human's performance due to the weight of the end-effector. Also, the damage rate is required to be reduced in future studies (Birrell et al., 2019). For the harvesting robot, the Mask-RCNN with ResNet-50 model was used to detect strawberries and achieved higher mean Intersection over Union (mIoU). But it is suggested that the real-time implementation can be improved by proposing a lightweight model and the sample size could also be increased to improve the performance (Yu et al., 2019). To perform two tasks simultaneously (detection of fruits and estimation of their ripeness), a CNN-based system was proposed in Halstead et al. (2018). An important task of fruit detection on a coffee crop was performed by UAV in which a simple ANN model was used and compared with well-known ML techniques like K-nearest neighbour and random forest classifier; ANN outperformed the ML techniques in terms of F-score (Carrijo et al., 2017). An Unmanned Ground Vehicle (UGV) system was deployed in an orchard for the detection of fruits and their yield was estimated by CNN, MLP, and WS algorithms. Future studies can be conducted to use the transfer learning technique and various labelling methods should be implemented to advance the performance (Bargoti & Underwood, 2017a, b). A comprehensive research was conducted for the detection of fruits in an orchard by UGV through Faster-RCNN with ZFNet and VGG-16. The secondary contribution of this research was the evaluation of the transfer learning method, the conclusion being that for the fruits detection task of the dataset used in that research, this approach was not very useful in terms of average precision, and the transfer learning strategy was suggested with variation/diversity in the dataset images (Bargoti & Underwood, 2017a, b). A robotic arm along with its grippers and recognition system was designed to harvest tomatoes; a YOLO model was used for the detection of tomatoes. It is to be noted that, following hardware design and obtaining good recognition and harvesting results, the applied YOLO method should be compared with other DL architectures like Faster-RCNN, SSD, etc. (Yeshmukhametov et al., 2019). A mobile robot was designed that consisted of robotic arms for the detection of tomatoes according to their maturity level. For that purpose, the MobileNet model with SSD architecture was selected due to its best performance in terms of classification accuracy after a comprehensive comparison of state-of-the-art DL architectures like YOLO-v3 and ResNet-152 with Faster-RCNN (Horng et al., 2019). An UR3 robotic arm was used for harvesting apples and Single Shot Multibox Detector (SSD) was used, although the implemented DL network achieved

more than 90% detection accuracy. But other efficient DL models (Faster RCNN or RFCN) should still be tested to further investigate the effectiveness of SSD for that task (Onishi et al., 2019). Another research article used a robotic manipulator to recognize seven vegetables by well-known DL feature extraction/detection architectures, but the recognition accuracy should be further increased by proposing some key modifications on the models applied in this research (Zheng et al., 2018).

From the explanation provided above, it is evident that several robotic manipulators have been proposed in many studies for the recognition/harvesting of fruits and vegetables, and various DL architectures have been implemented to perform these tasks in real-time. Nonetheless, only a few suggestions have been highlighted for the improvement in accuracy such as training the DL models by transfer learning technique and some modifications in famous DL models, etc.

Land Cover Classification

Land cover classification is a vast topic. Many studies have been conducted to classify land covers of various types to perform an overall analysis of one or more areas by DL-based techniques, specifically CNN architectures (Huang et al., 2018; Luus et al., 2015; Zhang, Harrison, et al., 2020; Zhang, Huang, et al., 2020). Some researchers used publicly available datasets by considering important lands of an area and performed classification studies by CNN models (Helber et al., 2019). In this review, only those studies were considered which incorporated agricultural land covers. For example, a single hidden layer neural network based on the extreme learning machine (ELM) method was proposed for this task and achieved comparable performance with a backpropagation neural network (BPNN) (Pal, 2009). An Unmanned Aircraft System (UAS) was implemented to classify land covers by Fully Convolutional Network (FCN), Support Vector Machines (SVM), Random Forest (RF), and Deep Convolutional Neural Network (DCNN) and concluded that DCNN and FCN have substantially higher accuracy than other classifiers, and authors suggested that multi-view data taken from the UAS can work with the DNN without needing a huge amount of training data (Liu, Abd-Elrahman, et al., 2018; Liu, Zhang, et al., 2018). To monitor forest cover, the CNN approach was adopted for the images taken from airborne and LiDAR. The weights can be optimized and other agricultural lands should be considered to prove the effectiveness of the method (Suzuki et al., 2018). The detection of citrus along with other crops' trees was performed by UAV through the implementation of a simple CNN model consisting of only one hidden layer (Csillik et al., 2018). A research letter was published to show the significance of the Deep Recurrent Neural Network (DRNN) for the task of land cover classification on the satellite images, found that one set of images achieved the highest accuracy by RF (LSTM) model while other datasets obtained best results by SVM (LSTM) model (Ienco et al., 2017). Another research was conducted to classify satellite images of 11 different crops' land by RNN predominantly Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU). This analysis was important to get the spatial information of the crops. It was suggested that multi-source data such as optical and SAR radar could be implemented in the future (Ndikumana et al., 2018). Another research was conducted to project the dynamics of forest cover by LSTM-based DL architecture, and relative explanatory variables to be included in future work and a robust deep learning model for different forest covers could be proposed (Ye et al., 2019). A study proposed the 3D-VGG model to show its effectiveness for the classification of crops' lands from the images taken by two satellites (Ji et al., 2018). In (Kussul et al.,

2017), a study presented the classification of various land cover types and crop types by the application of 1-D and 2-D CNNs for the first time specifically for the multisource satellite images, and the performance of CNNs was compared with RF and MLP. A study utilized hyperspectral images for the classification of various categories of natural vegetation and evaluated the performance of CNN, RF, and SVM. Moreover, accuracy can be increased by proposing an improved version of DL architecture (Guidici & Clark, 2017). A research was conducted by an airborne imaging system considering three kinds of datasets out of which two datasets were related to agricultural crops and the third dataset was related to various buildings; the applied CNN model was highly accurate to classify different crops for the first two locations (Song & Kim, 2017). Some well-known and successful DL models were trained and tested on wetland classes and found that the Inception ResNet-v2 model outperformed the other DL models including VGG, ResNet, Xception, DenseNet, Inception-v3 (Mahdianpari et al., 2018). The classification of 14 agricultural landscapes was performed by proposing six DL architectures and comparing the best CNN model with RF algorithm to prove its effectiveness in terms of its spatial feature extraction capability (Xie et al., 2019).

Various state-of-the-art DL models performed the task of agricultural land cover classification, especially CNNs and RNNs, which opens future research opportunities to manage various agricultural landscapes in a better way.

Overall Presentation of DL Algorithms for Agricultural Operations by Robots

The performance of DL/ML algorithms described in this section for the various agricultural operations is presented by bar plots in Figs. 6, 7, 8, and 9. In these plots, the DL/ML models are grouped by their respective research articles (denoted by D1, D2, D3, and so on) which are cited in Figs. 10, 11, 12, and 13 respectively. These figures are addressing the research questions mentioned in "Introduction" regarding the performance metrics/indicators, robotic platforms, and agricultural products, that have been commonly used during the implementation of deep learning architectures. It is also to be noted from the plots that some of the articles have shown the superiority of DL/ANN over traditional ML algorithms, like (Gutierrez et al., 2019) evaluated that RCNN outperformed the KNN model for the task of plant disease detection; in (Fan et al., 2018), the CNN achieved slightly better performance than SVM and RF for plant recognition purposes. To perform the crop/weed discrimination task, ResNet outperformed SVM and RF models (Bah et al., 2018), and, as described in (dos Santos Ferreira et al., 2017), a Convnet achieved better precision than RF. For the recognition of coffee, ANN obtained better results than KNN and RF (Carrijo et al., 2017). Moreover, several studies were conducted which proved the significance of DL models as compared to ML algorithms for the classification of agricultural land cover (Liu, Abd-Elrahman, et al., 2018; Liu, Zhang, et al., 2018), (Kussul et al., 2017), (Guidici & Clark, 2017) and (Xie et al., 2019). Similarly, from bar plots, it can also be observed that the deep learning-based image classification algorithms like AlexNet (Krizhevsky et al., 2012), ResNet (He et al., 2016), VGG (Simonyan & Zisserman, 2014), Xception (Chollet, 2017), MobileNet-v2 (Sandler et al., 2018) and object detection algorithms including Fast RCNN (Girshick, 2015), Faster RCNN (Ren et al., 2015), SSD (Liu et al., 2016), various versions of YOLO (You Only Look Once) models like YOLO-v1 (Redmon et al., 2016), YOLO-v2 (Redmon & Farhadi, 2017) and YOLO-v3 (Redmon & Farhadi, 2018) have been commonly used for various agricultural tasks. Therefore, upcoming research should incorporate any of the agricultural tasks by using successful deep learning models or proposing

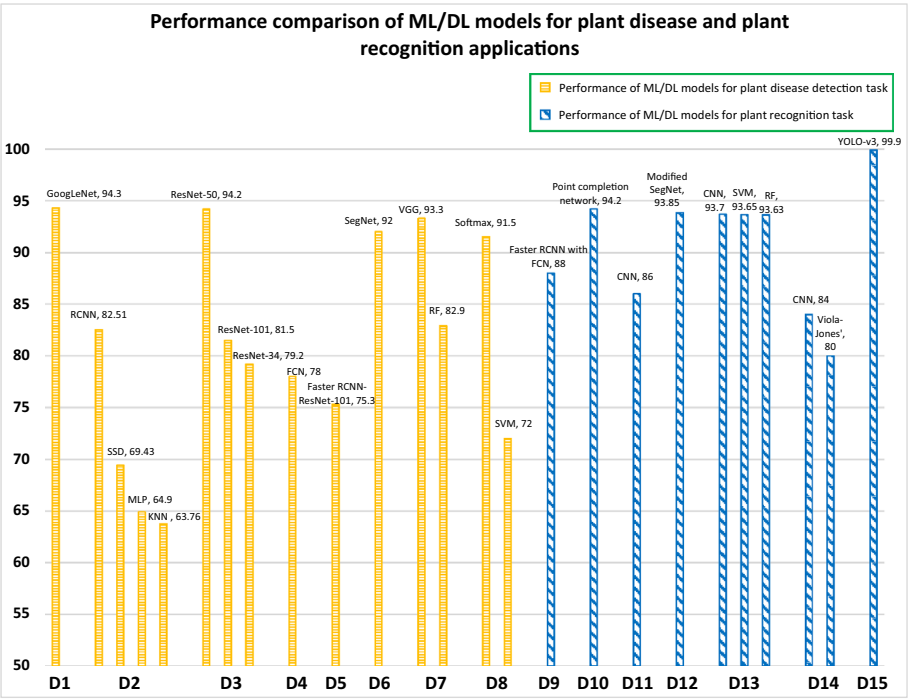


Fig. 6 Performance plots (in %) of ML/DL models used in robotic systems for plant disease detection (horizontal bars) and plant recognition (diagonal bars) tasks

modifications in the form of cascaded or hybrid versions, essential changes in convolutional layers, the number of filter, stride, etc. (Liu, Abd-Elrahman, et al., 2018; Liu, Zhang, et al., 2018; Singh et al., 2019; Zhang, Jia, et al., 2018; Zhang, Qiao, et al., 2018). Its performance should then be tested offline before its implementation on real robotic platforms. Some important research gaps/future directions from this section are provided in Table 3.

Conclusion and Future Directions

In this review, robotic solutions are presented for the major agricultural tasks by machine and deep learning algorithms. Moreover, the performance of machine learning models is summarized along with selected agricultural products and robotic platforms for certain agricultural operations. Furthermore, the performance plots are drawn to indicate the effectiveness of deep learning models for the respective agricultural tasks. From the plots, it can be concluded that the DL architectures outperformed traditional ML algorithms. Although significant developments have been observed in recent studies, still some important research gaps are identified to further advance the agricultural field of research.

A brief summary of prominent results to indicate the significance of the DL architectures as compared to the ML-based techniques applied through the robotic system for five selected agricultural applications, and few future works is presented as follows:

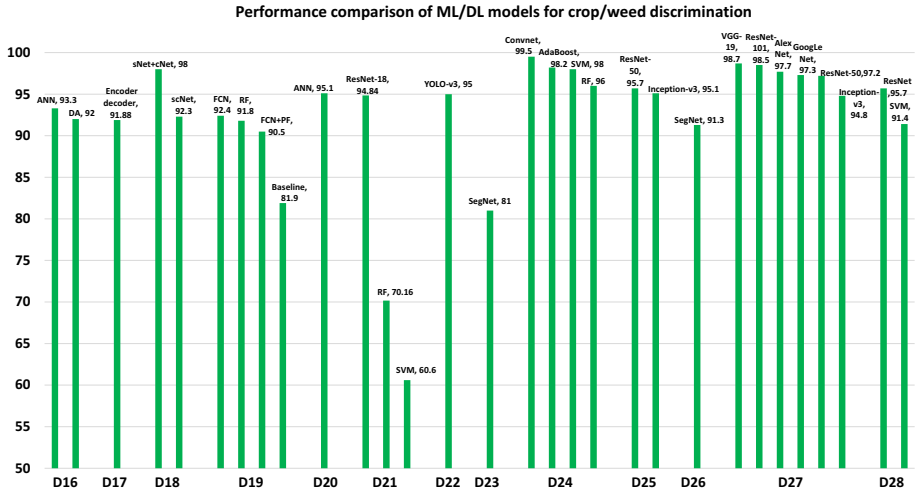


Fig. 7 Performance plots (in %) of ML/DL models used in robotic systems for crop/weed discrimination task

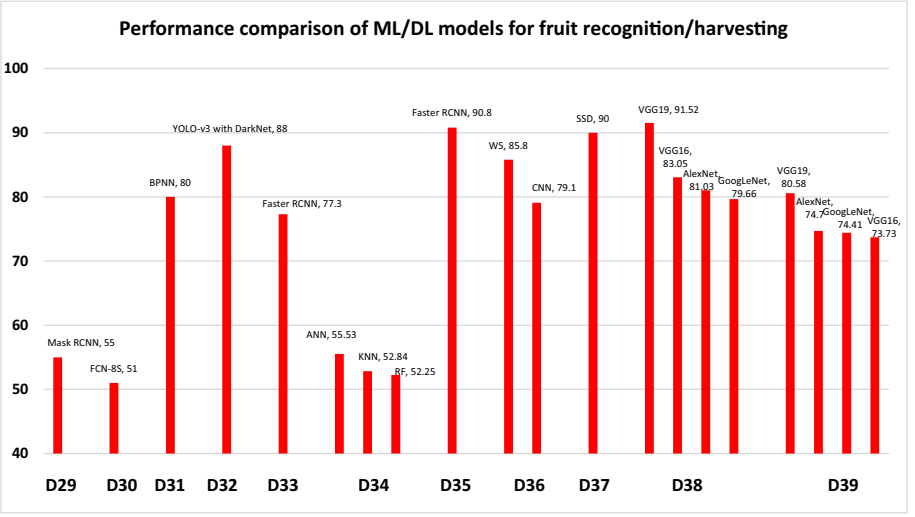


Fig. 8 Performance plots (in %) of ML/DL models used in robotic systems for fruit recognition and harvesting tasks

- Plant disease detection: RCNN achieved 82.51% detection rate, which was better than the other methods including SSD, MLP, and KNN with a difference of 13.08%, 17.61%, and 18.75%, respectively.
- Plant recognition: CNN attained 0.84 F-measure, which was greater than the Viola-Jones' method that achieved 0.80 F-measure.

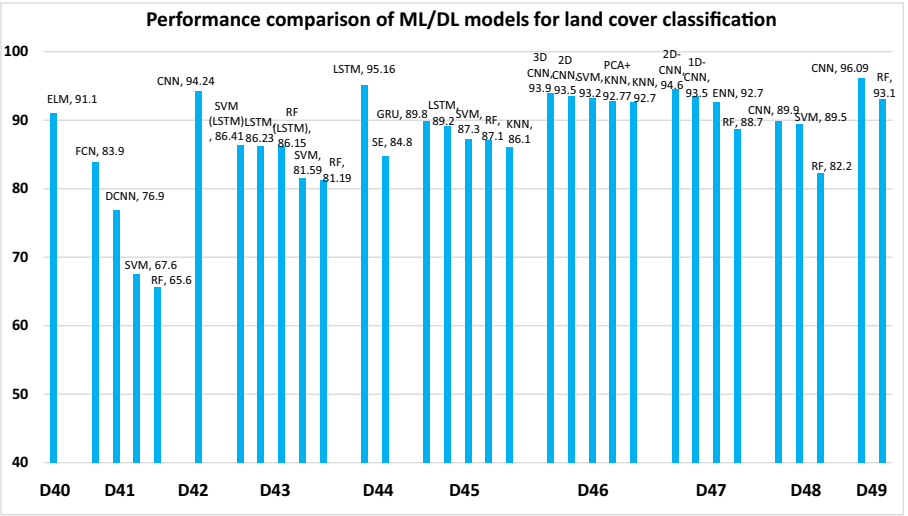


Fig. 9 Performance plots (in %) of ML/DL models used in robotic systems for the land cover classification

- Crop/weed discrimination: A well-known DL model named ResNet (94.84%) outperformed the traditional ML algorithms including SVM (60.6%) and RF (70.16%) in terms of area under the curve.
- Fruit recognition/harvesting: An ANN-based model achieved 0.5553 F-measure, which was slightly better than the ML models like KNN (0.5284) and RF (0.5255).
- Agricultural land cover classification: Three studies revealed that the performance of DL models was better than the ML-based techniques as listed below:
  - FCN got 83.9% overall accuracy, which was greater than DCNN (76.9%), SVM (67.6%), and RF (65.6%) models.
  - 2D-CNN attained a higher accuracy (94.6%) as compared to 1D-CNN (93.5%), ENN (92.7%), and RF (88.7%) models.
  - CNN (89.9%) performed better than SVM (89.5%) and RF models (82.2%) in terms of overall accuracy.
- Out of five major agricultural operations, plant disease detection and classification lack a comprehensive study. Although these agricultural tasks have been addressed by offline approaches in many research articles, these should be performed by a robotic manipulator/mobile robot through deep learning meta-architectures.
- After the successful application of DL algorithms for the detection/classification of plant disease by the robot, a combined effort by engineers and agronomists is required to implement a chemical spraying system that would apply fungicide/herbicide spray to the defected parts of the plant. It will be useful to reduce the cost of the crop protection system for agricultural farms.
- Most of the approaches were detected/classified disease in plant leaves, but the defects in other parts of the plant species should also be detected like stems/flowers.
- The adaptive nature of DL models should be utilized to show its automatic feature extraction capability for performing the various agricultural tasks by an efficient DL-based robot. For this purpose, the diversity in datasets must also be presented.

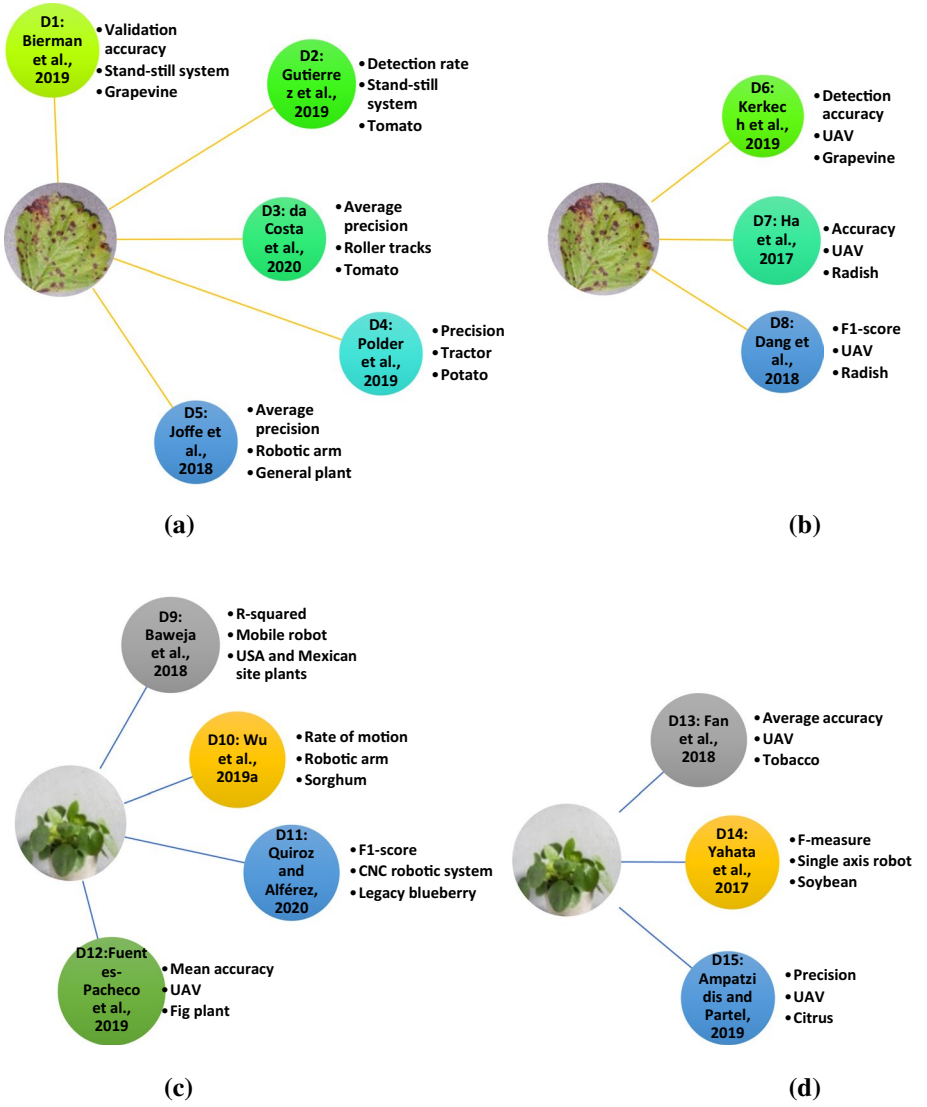


Fig. 10 The corresponding reference of research articles (D1–D15) linked to the bar plot (Fig. 6), and performance metrics along with robotic platforms and agricultural products; **a** and **b** present plant disease detection task, whereas, **c** and **d** present plant recognition task

- To improve the performance of various complex agricultural tasks, the modified/cascaded version of DL models should be proposed which can show their effectiveness by visualizing their convolutional layers.
- A multi-purpose robot should be designed to show its adaptive behaviour in a sense of its physical structure to perform various operations in a farm (a tractor is the best example of a robotic platform that can be used for various purposes like plowing, planting, and similar tasks).

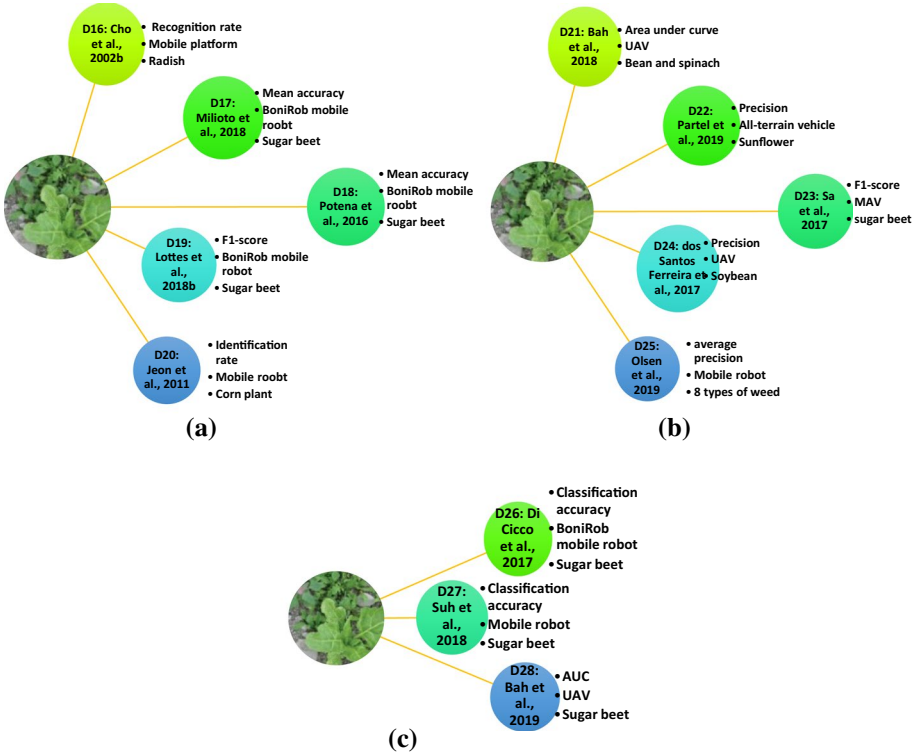


Fig. 11 The corresponding reference of research articles (D16-D28) linked to the bar plot (Fig. 7) for crop/weed discrimination task, and performance metrics along with robotic platforms and agricultural products

- To visualize the complex agricultural tasks like crop/weed discrimination and fruit detection, advanced visualization techniques such as saliency map should be applied.
- Some articles have previously presented to understand the factors affecting the performance of ML/DL algorithms for agricultural tasks, but a comprehensive study is still required for further development in agricultural automated systems.
- Improvement in land cover classification could be done by proposing an improved version of CNN/RNN.
- For better growth of agricultural products, an automated system should be proposed for the prediction of soil moisture content through robotic platforms.
- A recent topic like Internet of Robotic Things should also be deployed for agricultural purposes so that a new research area would be able to be explored.

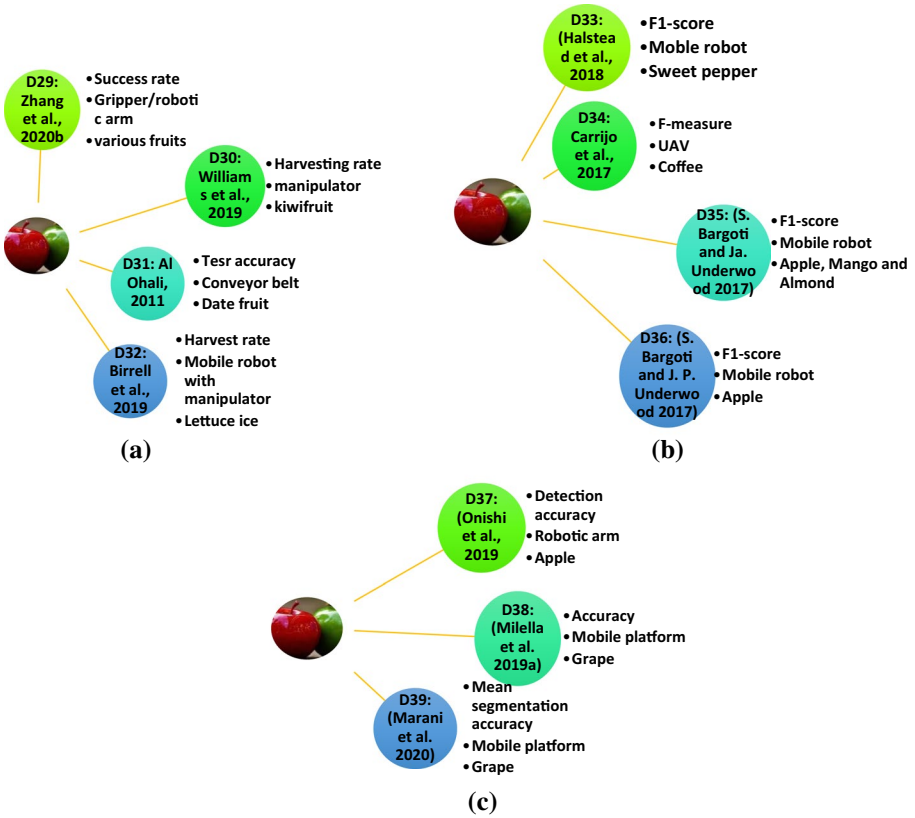


Fig. 12 The corresponding reference of research articles (D29-D39) linked to the bar plot (Fig. 8) for fruit recognition and harvesting tasks, and performance metrics along with robotic platforms and agricultural products

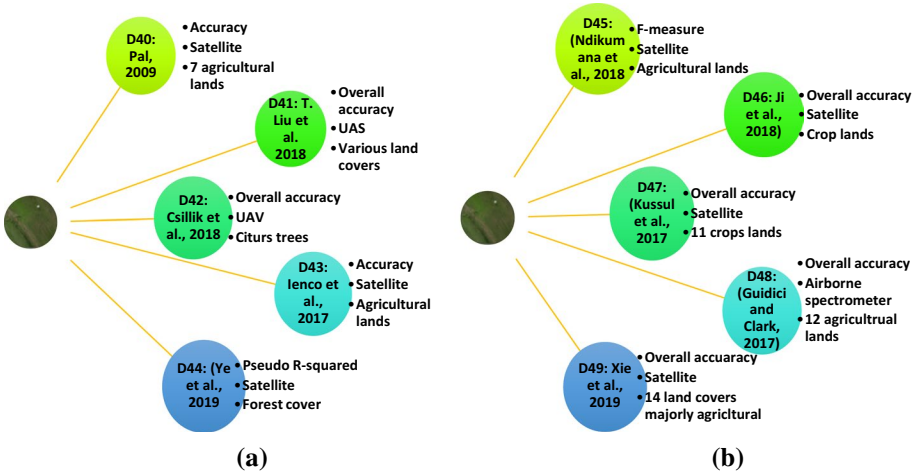


Fig. 13 The corresponding reference of research articles (D40-D49) linked to the bar plot (Fig. 9) for land cover classification task, and performance metrics along with robotic platforms and agricultural products

Table 3 Research gaps from some of the articles implemented DL architectures with their respective agricultural operations

Agricultural applications	Research gaps/future directions	Refs.
Plant disease detection	Data augmentation technique should be applied to generate diversity in the dataset The dataset could have included insects, pests to obtain diversity in the dataset	(Gutierrez et al., 2019)
	A deep autoencoder can be implemented to further improve the performance	(da Costa et al., 2020)
	The false detection could be reduced by applying DL models	(Kerkech et al., 2019)
	A combination of RGB and infrared images could improve overall performance The analysis should be done based on the severity of the disease	(Ha et al., 2017)
Plant recognition/stalk count	More accurate study for stalk width and stalk count could be attempted	(Baweja et al., 2018)
	The proposed method should also be tested on other agricultural crops	(Kitano et al., 2019)
	The proposed scheme can also be tested for pest detection The generative adversarial networks can also be deployed for generating synthetic images	(Quiroz & Alférez, 2020)
	The orthomosaic images could improve the system proposed in the paper	(Fuentes-Pacheco et al., 2019)
	Some other crops should also be considered in the future	(Fan et al., 2018)
	The diversity in the dataset could have achieved more improvement in the performance	(Ghosal et al., 2019)
	Seedpod counting could be one important future work	(Yahata et al., 2017)
Crop/weed discrimination	The improvement in background segmentation should be done by using multispectral images A graphical interface was suggested to generate an infestation. map The cost could be saved by the precise application of herbicide in the fields	(Bah et al., 2018)
	An algorithm could be applied that can vary the amount of chemical required to control the weed A smart sprayer should be deployed; its performance should be compared with the traditional sprayers	(Partel et al., 2019)
	A larger dataset could help in the improvement of performance	(Sa et al., 2017)
	The results were obtained under a controlled environment which leads to a research gap that can be filled by considering a real environment with a larger dataset	(dos Santos Ferreira et al., 2017)
	The use of NIR spectroscopy and hyperspectral imaging could be helpful to attain high accuracy	(Olsen et al., 2019)
	Multiple classes could also be considered to prove the robustness of the applied method	(Suh et al., 2018)

Table 3 (continued)

Agricultural applications	Research gaps/future directions	Refs.
Fruits/vegetables recognition and harvesting	Future work was given in the paper to increase accuracy to 70% and considering a greater degree-of-freedom which could increase cycle time	(Williams et al., 2019)
	An impact sensor should be deployed, and a feature distribution-based method should be introduced	(Al Ohali, 2011)
	The damage rate should be reduced	(Birrell et al., 2019)
	The real-time implementation can be improved by proposing a lightweight model, the samples size could also be increased to improve the performance	(Yu et al., 2019)
	A transfer learning strategy was suggested with variation/diversity in the dataset images	(Bargoti and Underwood, 2017a, b)
	The transfer learning technique and various labelling methods should be implemented	(Bargoti & Underwood, 2017a, b)
Land cover classification	It was suggested that the multi-view data taken from the UAS can result in DNN work without having a huge amount of training data	(Liu, Abd-Elrahman, et al., 2018; Liu, Zhang, et al., 2018)
	The weights can be optimized, and other agricultural lands should be considered to prove the effectiveness of the method	(Suzuki et al., 2018)
	The multi-source data such as optical and SAR radar can be implemented in the future	(Ndikumana et al., 2018)
	Relative explanatory variables and a robust deep learning model for different forest covers could be proposed	(Ye et al., 2019)
	The accuracy can be increased by proposing an improved version of DL architecture	(Guidici & Clark, 2017)

Funding This research was funded by the Ministry of Business, Innovation and Employment (MBIE), New Zealand, Science for Technological Innovation (SfTI) National Science Challenge.

Declarations

Conflict of interest The authors declare that they have no conflict of interests.

References

Adhikari, S. P., Yang, H., & Kim, H. (2019). Learning semantic graphics using convolutional encoder-decoder network for autonomous weeding in paddy field. *Frontiers in Plant Science, 10*, 1404

Al Ohali, Y. (2011). Computer vision based date fruit grading system: Design and implementation. *Journal of King Saud University-Computer and Information Sciences, 23*(1), 29–36

Alexandridis, T. K., Tamouridou, A. A., Pantazi, X. E., Lagopodi, A. L., Kashefi, J., Ovakoglou, G., et al. (2017). Novelty detection classifiers in weed mapping: Silybum marianum detection on UAV multispectral images. *Sensors, 17*(9), 2007

Altaheri, H., Alsulaiman, M., & Muhammad, G. (2019). Date fruit classification for robotic harvesting in a natural environment using deep learning. *IEEE Access, 7*, 117115–117133

Ampatzidis, Y., De Bellis, L., & Luvisi, A. (2017). iPathology: robotic applications and management of plants and plant diseases. *Sustainability, 9*(6), 1010

Ampatzidis, Y., & Partel, V. (2019). UAV-based high throughput phenotyping in citrus utilizing multispectral imaging and artificial intelligence. *Remote Sensing, 11*(4), 410

Arefi, A., & Motlagh, A. M. (2013). Development of an expert system based on wavelet transform and artificial neural networks for the ripe tomato harvesting robot. *Australian Journal of Crop Science, 7*(5), 699

Azouz, A. B., Esmonde, H., Corcoran, B., & O'Callaghan, E. (2015). Development of a teat sensing system for robotic milking by combining thermal imaging and stereovision technique. *Computers and Electronics in Agriculture, 110*, 162–170

Bac, C. W., van Henten, E. J., Hemming, J., & Edan, Y. (2014). Harvesting robots for high-value crops: State-of-the-art review and challenges ahead. *Journal of Field Robotics, 31*(6), 888–911

Bah, M. D., Hafiane, A., & Canals, R. (2018). Deep learning with unsupervised data labeling for weed detection in line crops in UAV images. *Remote Sensing, 10*(11), 1690

Bah, M. D., Hafiane, A., Canals, R., & Emile, B. (2019). Deep features and One-class classification with unsupervised data for weed detection in UAV images. In *Ninth International Conference on Image Processing Theory, Tools and Applications (IPTA), 2019* (pp. 1–5). Istanbul, Turkey: IEEE.

Bakhshipour, A., & Jafari, A. (2018). Evaluation of support vector machine and artificial neural networks in weed detection using shape features. *Computers and Electronics in Agriculture, 145*, 153–160

Bargoti, S., & Underwood, J. (2017a). Deep fruit detection in orchards. In *IEEE International Conference on Robotics and Automation (ICRA), 2017* (pp. 3626–3633). Marina Bay Sands, Singapore: IEEE.

Bargoti, S., & Underwood, J. P. (2017b). Image segmentation for fruit detection and yield estimation in apple orchards. *Journal of Field Robotics, 34*(6), 1039–1060

Barker, J., Sarathy, S., & July, A. (2016). DetectNet: Deep Neural Network for Object Detection in DIGITS. *Nvidia,(retrieved: 2016–11–30)*. Retrieved from https://devblogs.nvidia.com/parallelforall/detectnet-deep-neural-network-object-detection-digits.

Baweja, H. S., Parhar, T., Mirbod, O., & Nuske, S. Stalknet: A deep learning pipeline for high-throughput measurement of plant stalk count and stalk width. In *Field and Service Robotics, 2018* (pp. 271–284): Springer.

Behmann, J., Mahlein, A.-K., Rumpf, T., Römer, C., & Plümer, L. (2015). A review of advanced machine learning methods for the detection of biotic stress in precision crop protection. *Precision Agriculture, 16*(3), 239–260

Bierman, A., LaPlumm, T., Cadle-Davidson, L., Gadoury, D., Martinez, D., Sapkota, S., et al. (2019). A high-throughput phenotyping system using machine vision to quantify severity of grapevine powdery mildew. *Plant Phenomics, 2019*, 9209727

Birrell, S., Hughes, J., Cai, J. Y., & Iida, F. (2019). A field-tested robotic harvesting system for iceberg lettuce. *Journal of Field Robotics, 37*, 225–245

Brahimi, M., Arsenovic, M., Laraba, S., Sladojevic, S., Boukhalfa, K., & Moussaoui, A. (2018). Deep learning for plant diseases: detection and saliency map visualisation. In *Human and Machine Learning* (pp. 93–117): Springer.

Carrijo, G. L., Oliveira, D. E., de Assis, G. A., Carneiro, M. G., Guizilini, V. C., & Souza, J. R. (2017). Automatic detection of fruits in coffee crops from aerial images. In *Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR), 2017* (pp. 1–6). Curitiba, PR, Brazil: IEEE.

Chen, L.-C., Papandreou, G., Schroff, F., & Adam, H. (2017). Rethinking atrous convolution for semantic image segmentation. *arXiv preprint* arXiv:1706.05587.

Chen, Y., Lee, W. S., Gan, H., Peres, N., Fraisse, C., Zhang, Y., et al. (2019). Strawberry Yield Prediction Based on a Deep Neural Network Using High-Resolution Aerial Orthoimages. *Remote Sensing, 11*(13), 1584

Cheng, B., & Matson, E. T. (2015). A feature-based machine learning agent for automatic rice and weed discrimination. In *International Conference on Artificial Intelligence and Soft Computing, 2015* (pp. 517–527). Zakopane, Poland: Springer.

Cho, S., Chang, S., Kim, Y., & An, K. (2002). Development of a three-degrees-of-freedom robot for harvesting lettuce using machine vision and fuzzy logic control. *Biosystems Engineering, 82*(2), 143–149

Cho, S., Lee, D., & Jeong, J. (2002). AE—automation and emerging technologies: Weed–plant discrimination by machine vision and artificial neural network. *Biosystems Engineering, 83*(3), 275–280

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017* (pp. 1251–1258). Honolulu, HI, USA.

Csillik, O., Cherbini, J., Johnson, R., Lyons, A., & Kelly, M. (2018). Identification of citrus trees from unmanned aerial vehicle imagery using convolutional neural networks. *Drones, 2*(4), 39

da Costa, A. Z., Figueroa, H. E., & Fracarolli, J. A. (2020). Computer vision based detection of external defects on tomatoes using deep learning. *Biosystems Engineering, 190*, 131–144

Dang, L. M., Hassan, S. I., Suhyeon, I., Kumar Sangaiah, A., Mehmood, I., Rho, S., et al. (2018). UAV based wilt detection system via convolutional neural networks. *Sustainable Computing: Informatics and Systems*. https://doi.org/10.1016/j.suscom.2018.05.010

De-An, Z., Jidong, L., Wei, J., Ying, Z., & Yu, C. (2011). Design and control of an apple harvesting robot. *Biosystems Engineering, 110*(2), 112–122

Di Cicco, M., Potena, C., Grisetti, G., & Pretto, A. (2017). Automatic model based dataset generation for fast and accurate crop and weeds detection. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017* (pp. 5188–5195). Vancouver, BC, Canada: IEEE.

dos Santos Ferreira, A., Freitas, D. M., da Silva, G. G., Pistori, H., & Folhes, M. T. (2017). Weed detection in soybean crops using ConvNets. *Computers and Electronics in Agriculture, 143*, 314–324

dos Santos Ferreira, A., Freitas, D. M., da Silva, G. G., Pistori, H., & Folhes, M. T. (2019). Unsupervised deep learning and semi-automatic data labeling in weed discrimination. *Computers and Electronics in Agriculture, 165*, 104963

Duro, D. C., Franklin, S. E., & Dubé, M. G. (2012). A comparison of pixel-based and object-based image analysis with selected machine learning algorithms for the classification of agricultural landscapes using SPOT-5 HRG imagery. *Remote Sensing of Environment, 118*, 259–272

Dyrmann, M., Christiansen, P., & Midtiby, H. S. (2018). Estimation of plant species by classifying plants and leaves in combination. *Journal of Field Robotics, 35*(2), 202–212

Dyrmann, M., Jørgensen, R. N., & Midtiby, H. S. (2017). RoboWeedSupport-Detection of weed locations in leaf occluded cereal crops using a fully convolutional neural network. *Advances in Animal Biosciences, 8*(2), 842–847

Dyrmann, M., Karstoft, H., & Midtiby, H. S. (2016). Plant species classification using deep convolutional neural network. *Biosystems Engineering, 151*, 72–80

Ebrahimi, M., Khoshtaghaza, M., Minaei, S., & Jamshidi, B. (2017). Vision-based pest detection based on SVM classification method. *Computers and Electronics in Agriculture, 137*, 52–58

Eisavi, V., Homayouni, S., Yazdi, A. M., & Alimohammadi, A. (2015). Land cover mapping based on random forest classification of multitemporal spectral and thermal images. *Environmental Monitoring and Assessment, 187*(5), 291

Esgario, J. G., Krohling, R. A., & Ventura, J. A. (2020). Deep learning for classification and severity estimation of coffee leaf biotic stress. *Computers and Electronics in Agriculture, 169*, 105162

Fan, Z., Lu, J., Gong, M., Xie, H., & Goodman, E. D. (2018). Automatic tobacco plant detection in UAV images via deep neural networks. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 11*(3), 876–887

Feng, Q., Wang, X., Wang, G., & Li, Z. (2015). Design and test of tomatoes harvesting robot. In *IEEE International Conference on Information and Automation, 2015* (pp. 949–952). Lijiang, Yunnan, China: IEEE.

Fuentes-Pacheco, J., Torres-Olivares, J., Roman-Rangel, E., Cervantes, S., Juarez-Lopez, P., Hermosillo-Valadez, J., et al. (2019). Fig plant segmentation from aerial images using a deep convolutional encoder-decoder network. *Remote Sensing, 11*(10), 1157

Gao, J., Nuyttens, D., Lootens, P., He, Y., & Pieters, J. G. (2018). Recognising weeds in a maize crop using a random forest machine-learning algorithm and near-infrared snapshot mosaic hyperspectral imagery. *Biosystems Engineering, 170*, 39–50

Ghosal, S., Zheng, B., Chapman, S. C., Potgieter, A. B., Jordan, D. R., Wang, X., et al. (2019). A weakly supervised deep learning framework for sorghum head detection and counting. *Plant Phenomics, 2019*, 1525874

Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision, 2015* (pp. 1440–1448). Santiago, Chile: IEEE.

Gongal, A., Amatya, S., Karkee, M., Zhang, Q., & Lewis, K. (2015). Sensors and systems for fruit detection and localization: A review. *Computers and Electronics in Agriculture, 116*, 8–19

Guidici, D., & Clark, M. L. (2017). One-Dimensional convolutional neural network land-cover classification of multi-seasonal hyperspectral imagery in the San Francisco Bay Area, California. *Remote Sensing, 9*(6), 629

Gutierrez, A., Ansuategi, A., Susperregi, L., Tubío, C., Rankić, I., & Lenža, L. (2019). A benchmarking of learning strategies for pest detection and identification on tomato plants for autonomous scouting robots using internal databases. *Journal of Sensors*. https://doi.org/10.1155/2019/5219471

Gutiérrez, S., Fernández-Novales, J., Diago, M. P., & Tardaguila, J. (2018). On-the-go hyperspectral imaging under field conditions and machine learning for the classification of grapevine varieties. *Frontiers in Plant Science, 9*, 1102

Ha, J. G., Moon, H., Kwak, J. T., Hassan, S. I., Dang, M., Lee, O. N., et al. (2017). Deep convolutional neural network for classifying Fusarium wilt of radish from unmanned aerial vehicles. *Journal of Applied Remote Sensing, 11*(4), 042621

Hall, D., Dayoub, F., Kulk, J., & McCool, C. (2017). Towards unsupervised weed scouting for agricultural robotics. In *IEEE International Conference on Robotics and Automation (ICRA), 2017* (pp. 5223–5230). Marina Bay Sands, Singapore: IEEE.

Halstead, M., McCool, C., Denman, S., Perez, T., & Fookes, C. (2018). Fruit quantity and ripeness estimation using a robotic vision system. *IEEE Robotics and Automation Letters, 3*(4), 2995–3002

Haug, S., Michaels, A., Biber, P., & Ostermann, J. (2014). Plant classification system for crop/weed discrimination without segmentation. In *IEEE winter conference on applications of computer vision, 2014* (pp. 1142–1149). Steamboat Springs, CO, USA: IEEE.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition, 2016* (pp. 770–778). Las Vegas, NV, USA: IEEE.

Helber, P., Bischke, B., Dengel, A., & Borth, D. (2019). Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 12*(7), 2217–2226

Heremans, S., & Van Orshoven, J. (2015). Machine learning methods for sub-pixel land-cover classification in the spatially heterogeneous region of Flanders (Belgium): A multi-criteria comparison. *International Journal of Remote Sensing, 36*(11), 2934–2962

Horng, G.-J., Liu, M.-X., & Chen, C.-C. (2019). The smart image recognition mechanism for crop harvesting system in intelligent agriculture. *IEEE Sensors Journal, 20*, 2766–2781

Huang, B., Zhao, B., & Song, Y. (2018). Urban land-use mapping using a deep convolutional neural network with high spatial resolution multispectral remote sensing imagery. *Remote Sensing of Environment, 214*, 73–86

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition, 2017* (pp. 4700–4708). Honolulu, HI, USA: IEEE.

Huang, M., Tang, J., Yang, B., & Zhu, Q. (2016). Classification of maize seeds of different years based on hyperspectral imaging and model updating. *Computers and Electronics in Agriculture, 122*, 139–145

Huang, Y., Lan, Y., Thomson, S. J., Fang, A., Hoffmann, W. C., & Lacey, R. E. (2010). Development of soft computing and applications in agricultural and biological engineering. *Computers and Electronics in Agriculture, 71*(2), 107–127

Ienco, D., Gaetano, R., Dupaquier, C., & Maurel, P. (2017). Land cover classification via multitemporal spatial data by deep recurrent neural networks. *IEEE Geoscience and Remote Sensing Letters, 14*(10), 1685–1689

Ishimwe, R., Abutaleb, K., & Ahmed, F. (2014). Applications of thermal imaging in agriculture: A review. *Advances in Remote Sensing, 3*(03), 128

Jeon, H. Y., Tian, L. F., & Zhu, H. (2011). Robust crop and weed segmentation under uncontrolled outdoor illumination. *Sensors, 11*(6), 6270–6283

Jha, K., Doshi, A., Patel, P., & Shah, M. (2019). A comprehensive review on automation in agriculture using artificial intelligence. *Artificial Intelligence in Agriculture, 2*, 1–12

Ji, S., Zhang, C., Xu, A., Shi, Y., & Duan, Y. (2018). 3D convolutional neural networks for crop classification with multi-temporal remote sensing images. *Remote Sensing, 10*(1), 75

Ji, W., Zhao, D., Cheng, F., Xu, B., Zhang, Y., & Wang, J. (2012). Automatic recognition vision system guided for apple harvesting robot. *Computers & Electrical Engineering, 38*(5), 1186–1195

Jia, W., Mou, S., Wang, J., Liu, X., Zheng, Y., Lian, J., et al. (2020). Fruit recognition based on pulse coupled neural network and genetic Elman algorithm application in apple harvesting robot. *International Journal of Advanced Robotic Systems, 17*(1), 1729881419897473

Jodas, D. S., Marranghello, N., Pereira, A. S., & Guido, R. C. (2013). Comparing support vector machines and artificial neural networks in the recognition of steering angle for driving of mobile robots through paths in plantations. *Procedia Computer Science, 18*, 240–249

Joffe, B., Ahlin, K., Hu, A.-P., & McMurray, G. (2018). Vision-guided robotic leaf picking. *EasyChair Preprint, 250*, 1–6

Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture, 147*, 70–90

Kazerouni, M. F., Saeed, N. T. M., & Kuhnert, K.-D. (2019). Fully-automatic natural plant recognition system using deep neural network for dynamic outdoor environments. *SN Applied Sciences, 1*(7), 756

Kerkech, M., Hafiane, A., & Canals, R. (2019). Vine disease detection in UAV multispectral images with deep learning segmentation approach. *arXiv* preprint arXiv:1912.05281.

Kitano, B. T., Mendes, C. C., Geus, A. R., Oliveira, H. C., & Souza, J. R. (2019). Corn Plant Counting Using Deep Learning and UAV Images. *IEEE Geoscience and Remote Sensing Letters*.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems, 2012* (pp. 1097–1105).

Kurtulmus, F., Lee, W. S., & Vardar, A. (2011). Green citrus detection using 'eigenfruit', color and circular Gabor texture features under natural outdoor conditions. *Computers and Electronics in Agriculture, 78*(2), 140–149

Kussul, N., Lavreniuk, M., Skakun, S., & Shelestov, A. (2017). Deep learning classification of land cover and crop types using remote sensing data. *IEEE Geoscience and Remote Sensing Letters, 14*(5), 778–782

Kusumam, K., Krajník, T., Pearson, S., Cielniak, G., & Duckett, T. (2016). Can you pick a broccoli? 3D-vision based detection and localisation of broccoli heads in the field. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016* (pp. 646–651). Daejeon Convention Center (DCC), Daejeon, South Korea: IEEE.

Kusumam, K., Krajník, T., Pearson, S., Duckett, T., & Cielniak, G. (2017). 3D-vision based detection, localization, and sizing of broccoli heads in the field. *Journal of Field Robotics, 34*(8), 1505–1518

Kwak, G.-H., & Park, N.-W. (2019). Impact of texture information on crop classification with machine learning and UAV images. *Applied Sciences, 9*(4), 643

Lee, S. H., Chan, C. S., Mayo, S. J., & Remagnino, P. (2017). How deep learning extracts and learns leaf features for plant classification. *Pattern Recognition, 71*, 1–13

Lee, S. H., Chan, C. S., & Remagnino, P. (2018). Multi-organ plant classification based on convolutional and recurrent neural networks. *IEEE Transactions on Image Processing, 27*(9), 4287–4301

Li, P., Lee, S.-H., & Hsu, H.-Y. (2011). Review on fruit harvesting method for potential use of automatic fruit harvesting systems. *Procedia Engineering, 23*, 351–366

Li, Y., Wang, H., Dang, L. M., Sadeghi-Niaraki, A., & Moon, H. (2020). Crop pest recognition in natural scenes using convolutional neural networks. *Computers and Electronics in Agriculture, 169*, 105174

Liu, B., Zhang, Y., He, D., & Li, Y. (2018). Identification of apple leaf diseases based on deep convolutional neural networks. *Symmetry, 10*(1), 11

Liu, G., Mao, S., & Kim, J. H. (2019). A mature-tomato detection algorithm using machine learning and color analysis. *Sensors, 19*(9), 2023

Liu, J., Pi, J., & Xia, L. (2019). A novel and high precision tomato maturity recognition algorithm based on multi-level deep residual network. *Multimedia Tools and Applications, 79*, 9403–9417

Liu, T., Abd-Elrahman, A., Morton, J., & Wilhelm, V. L. (2018). Comparing fully convolutional networks, random forest, support vector machine, and patch-based deep convolutional neural networks for object-based wetland mapping using images from small unmanned aircraft system. *GIScience & Remote Sensing, 55*(2), 243–264

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., et al. Ssd: Single shot multibox detector. In *European conference on computer vision, 2016* (pp. 21–37). Amsterdam, Netherlands: Springer.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition, 2015* (pp. 3431–3440). Boston, MA, USA: IEEE.

Lottes, P., Behley, J., Chebrolu, N., Milioto, A., & Stachniss, C. (2018). Joint stem detection and crop-weed classification for plant-specific treatment in precision farming. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018a* (pp. 8233–8238). Madrid, Spain: IEEE.

Lottes, P., Behley, J., Milioto, A., & Stachniss, C. (2018). Fully convolutional networks with sequential information for robust crop and weed detection in precision farming. *IEEE Robotics and Automation Letters, 3*(4), 2870–2877

Lottes, P., Hoeferlin, M., Sander, S., Müter, M., Schulze, P., & Stachniss, L. C. (2016). An effective classification system for separating sugar beets and weeds for precision farming applications. In *IEEE International Conference on Robotics and Automation (ICRA), 2016* (pp. 5157–5163). Stockholm Waterfront Congress Centre, Stockholm, Sweden: IEEE.

Lottes, P., Khanna, R., Pfeifer, J., Siegwart, R., & Stachniss, C. (2017). UAV-based crop and weed classification for smart farming. In *IEEE International Conference on Robotics and Automation (ICRA), 2017* (pp. 3024–3031). Marina Bay Sands, Singapore: IEEE.

Luus, F. P., Salmon, B. P., Van den Bergh, F., & Maharaj, B. T. J. (2015). Multiview deep learning for land-use classification. *IEEE Geoscience and Remote Sensing Letters, 12*(12), 2448–2452

Mahdianpari, M., Salehi, B., Rezaee, M., Mohammadimanesh, F., & Zhang, Y. (2018). Very deep convolutional neural networks for complex land cover mapping using multispectral remote sensing imagery. *Remote Sensing, 10*(7), 1119

Mahlein, A., Kuska, M., Thomas, S., Bohnenkamp, D., Alisaac, E., Behmann, J., et al. (2017). Plant disease detection by hyperspectral imaging: From the lab to the field. *Advances in Animal Biosciences, 8*(2), 238–243

Mao, S., Li, Y., Ma, Y., Zhang, B., Zhou, J., & Wang, K. (2020). Automatic cucumber recognition algorithm for harvesting robots in the natural environment using deep learning and multi-feature fusion. *Computers and Electronics in Agriculture, 170*, 105254

Marani, R., Milella, A., Petitti, A., & Reina, G. (2020). Deep neural networks for grape bunch segmentation in natural images from a consumer-grade camera. *Precision Agriculture, 22*, 387–413

McCool, C., Perez, T., & Upcroft, B. (2017). Mixtures of lightweight deep convolutional neural networks: Applied to agricultural robotics. *IEEE Robotics and Automation Letters, 2*(3), 1344–1351

Milella, A., Marani, R., Petitti, A., & Reina, G. (2019). In-field high throughput grapevine phenotyping with a consumer-grade depth camera. *Computers and Electronics in Agriculture, 156*, 293–306

Milella, A., Reina, G., & Nielsen, M. (2019). A multi-sensor robotic platform for ground mapping and estimation beyond the visible spectrum. *Precision Agriculture, 20*(2), 423–444

Milella, A., Reina, G., Underwood, J., & Douillard, B. Combining radar and vision for self-supervised ground segmentation in outdoor environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2011* (pp. 255–260). San Francisco, CA, USA: IEEE.

Milioto, A., Lottes, P., & Stachniss, C. (2018). Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in CNNs. In *IEEE International Conference on Robotics and Automation (ICRA), 2018* (pp. 2229–2235). Brisbane, Australia: IEEE.

Narvaez, F. Y., Reina, G., Torres-Torriti, M., Kantor, G., & Cheein, F. A. (2017). A survey of ranging and imaging techniques for precision agriculture phenotyping. *IEEE/ASME Transactions on Mechatronics, 22*(6), 2428–2439

Nashat, S., Abdullah, A., Aramvith, S., & Abdullah, M. (2011). Support vector machine approach to real-time inspection of biscuits on moving conveyor belt. *Computers and Electronics in Agriculture, 75*(1), 147–158

Ndikumana, E., Ho Tong Minh, D., Baghdadi, N., Courault, D., & Hossard, L. (2018). Deep recurrent neural network for agricultural classification using multitemporal SAR Sentinel-1 for Camargue. *France. Remote Sensing, 10*(8), 1217

Nkemelu, D. K., Omeiza, D., & Lubalo, N. (2018). Deep convolutional neural network for plant seedlings classification. *arXiv preprint* arXiv:1811.08404.

Ok, A. O., Akar, O., & Gungor, O. (2012). Evaluation of random forest method for agricultural crop classification. *European Journal of Remote Sensing, 45*(1), 421–432

Olsen, A., Konovalov, D. A., Philippa, B., Ridd, P., Wood, J. C., Johns, J., et al. (2019). DeepWeeds: A multiclass weed species image dataset for deep learning. *Scientific Reports, 9*(1), 1–12

Onishi, Y., Yoshida, T., Kurita, H., Fukao, T., Arihara, H., & Iwai, A. (2019). An automated fruit harvesting robot by using deep learning. *ROBOMECH Journal, 6*(1), 13

Padarian, J., Minasny, B., & McBratney, A. (2019). Using deep learning to predict soil properties from regional spectral data. *Geoderma Regional, 16*, e00198

Pal, M. (2009). Extreme-learning-machine-based land cover classification. *International Journal of Remote Sensing, 30*(14), 3835–3841

Pantazi, X. E., Moshou, D., & Tamouridou, A. A. (2019). Automated leaf disease detection in different crop species through image features analysis and One Class Classifiers. *Computers and Electronics in Agriculture, 156*, 96–104

Partel, V., Kakarla, S. C., & Ampatzidis, Y. (2019). Development and evaluation of a low-cost and smart technology for precision weed management utilizing artificial intelligence. *Computers and Electronics in Agriculture, 157*, 339–350

Patrício, D. I., & Rieder, R. (2018). Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review. *Computers and Electronics in Agriculture, 153*, 69–81

Patrick, A., Pelham, S., Culbreath, A., Holbrook, C. C., De Godoy, I. J., & Li, C. (2017). High throughput phenotyping of tomato spot wilt disease in peanuts using unmanned aerial systems and multispectral imaging. *IEEE Instrumentation & Measurement Magazine, 20*(3), 4–12

Peña, J. M., Gutiérrez, P. A., Hervás-Martínez, C., Six, J., Plant, R. E., & López-Granados, F. (2014). Object-based image classification of summer crops with machine learning methods. *Remote Sensing, 6*(6), 5019–5041

Polder, G., Blok, P. M., de Villiers, H., van der Wolf, J. M., & Kamp, J. (2019). Potato virus y detection in seed potatoes using deep learning on hyperspectral images. *Frontiers in Plant Science, 10*, 209

Potena, C., Nardi, D., & Pretto, A. (2016). Fast and accurate crop and weed identification with summarized train sets for precision agriculture. In *International Conference on Intelligent Autonomous Systems, 2016* (pp. 105–121). Shanghai, China: Springer.

Pourazar, H., Samadzadegan, F., & Javan, F. D. (2019). Aerial Multispectral Imagery for Plant Disease Detection; Radiometric Calibration Necessity Assessment.

Quiroz, I. A., & Alférez, G. H. (2020). Image recognition of Legacy blueberries in a Chilean smart farm through deep learning. *Computers and Electronics in Agriculture, 168*, 105044

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition, 2016* (pp. 779–788). Las Vegas, NV, USA: IEEE.

Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition, 2017* (pp. 7263–7271). Honolulu, HI, USA: IEEE.

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint* arXiv:1804.02767.

Rehman, T. U., Mahmud, M. S., Chang, Y. K., Jin, J., & Shin, J. (2019). Current and future applications of statistical machine learning algorithms for agricultural machine vision systems. *Computers and Electronics in Agriculture, 156*, 585–605

Reina, G., Milella, A., & Galati, R. (2017). Terrain assessment for precision agriculture using vehicle dynamic modelling. *Biosystems Engineering, 162*, 124–139

Reina, G., Milella, A., Rouveure, R., Nielsen, M., Worst, R., & Blas, M. R. (2016). Ambient awareness for agricultural robotic vehicles. *Biosystems Engineering, 146*, 114–132

Ren, S., He, K., Girshick, R., & Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems, 2015* (pp. 91–99).

Rodriguez-Galiano, V. F., Ghimire, B., Rogan, J., Chica-Olmo, M., & Rigol-Sanchez, J. P. (2012). An assessment of the effectiveness of a random forest classifier for land-cover classification. *ISPRS Journal of Photogrammetry and Remote Sensing, 67*, 93–104

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention, 2015* (pp. 234–241). Munich, Germany: Springer.

Sa, I., Chen, Z., Popović, M., Khanna, R., Liebisch, F., Nieto, J., et al. (2017). weednet: Dense semantic weed classification using multispectral images and mav for smart farming. *IEEE Robotics and Automation Letters, 3*(1), 588–595

Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., & McCool, C. (2016). Deepfruits: A fruit detection system using deep neural networks. *Sensors, 16*(8), 1222

Saleem, M. H., Potgieter, J., & Arif, K. M. (2019). Plant disease detection and classification by deep learning. *Plants, 8*(11), 468

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition, 2018* (pp. 4510–4520). Salt Lake City, UT, USA: IEEE.

Santos, L., Santos, F. N., Oliveira, P. M., & Shinde, P. Deep learning applications in agriculture: A short review. In *Fourth Iberian Robotics conference, 2019* (pp. 139–151). Porto, Portugal: Springer.

Sarkar, S. K., Das, J., Ehsani, R., & Kumar, V. (2016). Towards autonomous phytopathology: Outcomes and challenges of citrus greening disease detection through close-range remote sensing. In *IEEE*

International Conference on Robotics and Automation (ICRA), 2016 (pp. 5143–5148). Stockholm, Sweden: IEEE.

Sengupta, S., & Lee, W. S. (2014). Identification and determination of the number of immature green citrus fruit in a canopy under different ambient light conditions. *Biosystems Engineering, 117*, 51–61

Shao, Y., & Lunetta, R. S. (2012). Comparison of support vector machine, neural network, and CART algorithms for the land-cover classification using limited training data points. *ISPRS Journal of Photogrammetry and Remote Sensing, 70*, 78–87

Sharif, M., Khan, M. A., Iqbal, Z., Azam, M. F., Lali, M. I. U., & Javed, M. Y. (2018). Detection and classification of citrus diseases in agriculture based on optimized weighted segmentation and feature selection. *Computers and Electronics in Agriculture, 150*, 220–234

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint* arXiv:1409.1556.

Singh, U. P., Chouhan, S. S., Jain, S., & Jain, S. (2019). Multilayer convolution neural network for the classification of mango leaves infected by anthracnose disease. *IEEE Access, 7*, 43721–43729

Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., & Stefanovic, D. (2016). Deep neural networks based recognition of plant diseases by leaf image classification. *Computational Intelligence and Neuroscience*. https://doi.org/10.1155/2016/3289801

Slaughter, D. C., Giles, D. K., Fennimore, S. A., & Smith, R. F. (2008). Multispectral machine vision identification of lettuce and weed seedlings for automated weed control. *Weed Technology, 22*(2), 378–384

Song, A., & Kim, Y. (2017). Deep learning-based hyperspectral image classification with application to environmental geographic information systems. *Korean Journal of Remote Sensing, 33*, 1061–1073

Sonobe, R., Tani, H., Wang, X., Kobayashi, N., & Shimamura, H. (2014). Random forest classification of crop type using multi-temporal TerraSAR-X dual-polarimetric data. *Remote Sensing Letters, 5*(2), 157–164

Suh, H. K., Ijsselmuiden, J., Hofstee, J. W., & van Henten, E. J. (2018). Transfer learning for the classification of sugar beet and volunteer potato under field conditions. *Biosystems Engineering, 174*, 50–65

Sujaritha, M., Annadurai, S., Satheeshkumar, J., Sharan, S. K., & Mahesh, L. (2017). Weed detecting robot in sugarcane fields using fuzzy real time classifier. *Computers and Electronics in Agriculture, 134*, 160–171

Suzuki, K., Rin, U., Maeda, Y., & Takeda, H. (2018). Forest cover classification using geospatial multimodal DaTA. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences, 42*(2), 1091–1096

Tao, Y., & Zhou, J. (2017). Automatic apple recognition based on the fusion of color and 3D feature for robotic fruit picking. *Computers and Electronics in Agriculture, 142*, 388–396

Tellaeche, A., Pajares, G., Burgos-Artizzu, X. P., & Ribeiro, A. (2011). A computer vision approach for weeds identification through Support Vector Machines. *Applied Soft Computing, 11*(1), 908–915

Thanh Noi, P., & Kappas, M. (2018). Comparison of random forest, k-nearest neighbor, and support vector machine classifiers for land cover classification using Sentinel-2 imagery. *Sensors, 18*(1), 18

Ubbens, J. R., & Stavness, I. (2017). Deep plant phenomics: a deep learning platform for complex plant phenotyping tasks. *Frontiers in plant science, 8*, 1190

Virnodkar, S. S., Pachghare, V. K., Patil, V., & Jha, S. K. (2020). Remote sensing and machine learning for crop water stress determination in various crops: A critical review. *Precision Agriculture, 21*, 1121–1155

Wan, S., & Goudos, S. (2020). Faster R-CNN for multi-class fruit detection using a robotic vision system. *Computer Networks, 168*, 107036

Wang, A., Zhang, W., & Wei, X. (2019). A review on weed detection using ground-based machine vision and image processing techniques. *Computers and Electronics in Agriculture, 158*, 226–240

Wang, D., Vinson, R., Holmes, M., Seibel, G., Bechar, A., Nof, S., et al. (2019). Early detection of tomato spotted wilt virus by hyperspectral imaging and outlier removal auxiliary classifier generative adversarial nets (OR-AC-GAN). *Scientific Reports, 9*(1), 4377

Wei, X., Jia, K., Lan, J., Li, Y., Zeng, Y., & Wang, C. (2014). Automatic method of fruit object extraction under complex agricultural background for vision system of fruit picking robot. *Optik-International Journal for Light and Electron Optics, 125*(19), 5684–5689

Weiss, U., Biber, P., Laible, S., Bohlmann, K., & Zell, A. (2010). Plant species classification using a 3D LIDAR sensor and machine learning. In *Ninth International Conference on Machine Learning and Applications, 2010* (pp. 339-345). Washington, DC, USA: IEEE.

Williams, H. A., Jones, M. H., Nejati, M., Seabright, M. J., Bell, J., Penhall, N. D., et al. (2019). Robotic kiwifruit harvesting using machine vision, convolutional neural networks, and robotic arms. *Biosystems Engineering, 181*, 140–156

Wolfert, S., Ge, L., Verdouw, C., & Bogaardt, M.-J. (2017). Big data in smart farming: A review. *Agricultural Systems, 153*, 69–80

Wspanialy, P., & Moussa, M. (2016). Early powdery mildew detection system for application in greenhouse automation. *Computers and Electronics in Agriculture, 127*, 487–494

Wu, C., Zeng, R., Pan, J., Wang, C. C., & Liu, Y.-J. (2019). Plant phenotyping by deep-learning-based planner for multi-robots. *IEEE Robotics and Automation Letters, 4*(4), 3113–3120

Wu, J., Zhang, B., Zhou, J., Xiong, Y., Gu, B., & Yang, X. (2019). Automatic recognition of ripening tomatoes by combining multi-feature fusion with a bi-layer classification strategy for harvesting robots. *Sensors, 19*(3), 612

Xie, B., Zhang, H. K., & Xue, J. (2019). Deep convolutional neural network for mapping smallholder agriculture using high spatial resolution satellite image. *Sensors, 19*(10), 2398

Xie, H., Fan, Z., Li, W., Rong, Y., Xiao, Y., & Zhao, L. (2016). Tobacco plant recognizing and counting based on svm. In *International Conference on Industrial Informatics-Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII), 2016* (pp. 109–113). Wuhan, China: IEEE.

Yahata, S., Onishi, T., Yamaguchi, K., Ozawa, S., Kitazono, J., Ohkawa, T., et al. (2017). A hybrid machine learning approach to automatic plant phenotyping for smart agriculture. In *International Joint Conference on Neural Networks (IJCNN), 2017* (pp. 1787–1793). Anchorage, Alaska: IEEE.

Yamamoto, K., Guo, W., Yoshioka, Y., & Ninomiya, S. (2014). On plant detection of intact tomato fruits using image analysis and machine learning methods. *Sensors, 14*(7), 12191–12206

Ye, L., Gao, L., Marcos-Martinez, R., Mallants, D., & Bryan, B. A. (2019). Projecting Australia's forest cover dynamics and exploring influential factors using deep learning. *Environmental Modelling & Software, 119*, 407–417

Yeshmukhametov, A., Koganezawa, K., Buribayev, Z., Amirgaliyev, Y., & Yamamoto, Y. (2019). Development of Continuum Robot Arm and Gripper for Harvesting Cherry Tomatoes.

Yu, Y., Zhang, K., Yang, L., & Zhang, D. (2019). Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN. *Computers and Electronics in Agriculture, 163*, 104846

Zhang, C., Harrison, P. A., Pan, X., Li, H., Sargent, I., & Atkinson, P. M. (2020). Scale Sequence Joint Deep Learning (SS-JDL) for land use and land cover classification. *Remote Sensing of Environment, 237*, 111593

Zhang, C., & Kovacs, J. M. (2012). The application of small unmanned aerial systems for precision agriculture: A review. *Precision Agriculture, 13*(6), 693–712

Zhang, L., Gui, G., Khattak, A. M., Wang, M., Gao, W., & Jia, J. (2019). Multi-task cascaded convolutional networks based intelligent fruit detection for designing automated robot. *IEEE Access, 7*, 56028–56038

Zhang, L., Jia, J., Gui, G., Hao, X., Gao, W., & Wang, M. (2018). Deep learning based improved classification system for designing tomato harvesting robot. *IEEE Access, 6*, 67940–67950

Zhang, T., Huang, Z., You, W., Lin, J., Tang, X., & Huang, H. (2020). An autonomous fruit and vegetable harvester with a low-cost gripper using a 3D sesnsor. *Sensors, 20*(1), 93

Zhang, X., Qiao, Y., Meng, F., Fan, C., & Zhang, M. (2018). Identification of maize leaf diseases using improved deep convolutional neural networks. *IEEE Access, 6*, 30370–30377

Zhao, Y., Gong, L., Huang, Y., & Liu, C. (2016). A review of key techniques of vision-based control for harvesting robot. *Computers and Electronics in Agriculture, 127*, 311–323

Zhao, Y., Gong, L., Zhou, B., Huang, Y., & Liu, C. (2016). Detecting tomatoes in greenhouse scenes by combining AdaBoost classifier and colour analysis. *Biosystems Engineering, 148*, 127–137

Zheng, Y.-Y., Kong, J.-L., Jin, X.-B., Su, T.-L., Nie, M.-J., & Bai, Y.-T. (2018). Real-Time Vegetables Recognition System based on Deep Learning Network for Agricultural Robots. In *Chinese Automation Congress (CAC), 2018* (pp. 2223–2228). Xi'an, China: IEEE.

Zujevs, A., Osadcuks, V., & Ahrendt, P. (2015). Trends in robotic sensor technologies for fruit harvesting: 2010–2015. *Procedia Computer Science, 77*, 227–233

Precision Agriculture (2021) 22:2092–2094
https://doi.org/10.1007/s11119-021-09824-9

CORRECTION

Correction to: Automation in Agriculture by Machine and Deep Learning Techniques: A Review of Recent Developments

Muhammad Hammad Saleem[1] · Johan Potgieter[2] · Khalid Mahmood Arif[1]

Published online: 21 June 2021

Correction to: Precision Agriculture

https://doi.org/10.1007/s11119-021-09806-x

The original version of this article unfortunately contained mistakes. The presentation of Figs. 6, 7, 8, 9 was incorrect. The correct versions are given below.

The original article has also been corrected.

The original article can be found online at https://doi.org/10.1007/s11119-021-09806-x.

✉ Khalid Mahmood Arif
k.arif@massey.ac.nz

[1] Department of Mechanical and Electrical Engineering, School of Food and Advanced Technology, Massey University, Auckland 0632, New Zealand

[2] Massey Agritech Partnership Research Centre, School of Food and Advanced Technology, Massey University, Palmerston North 4442, New Zealand

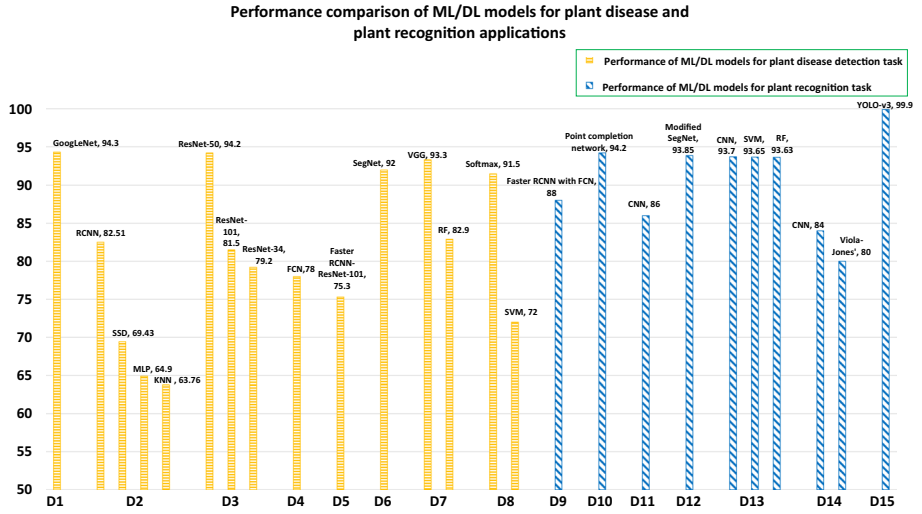


Fig. 6 Performance plots (in %) of ML/DL models used in robotic systems for plant disease detection (horizontal bars) and plant recognition (diagonal bars) tasks

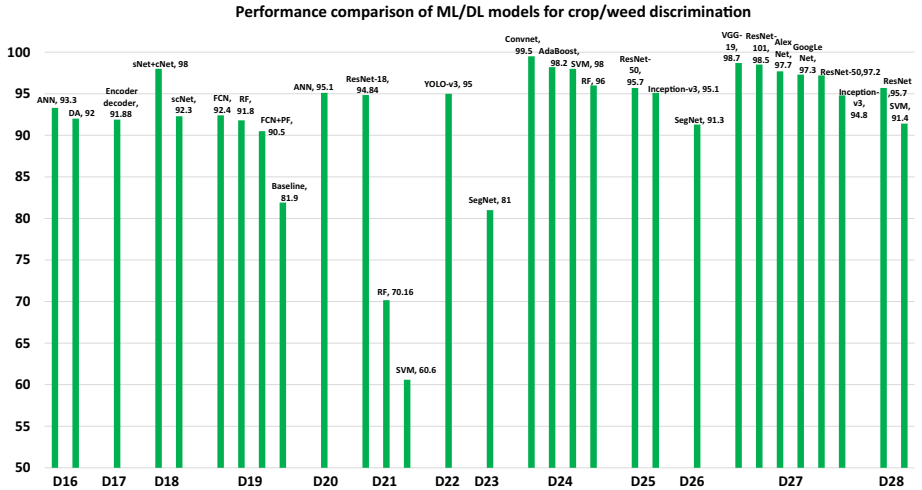


Fig. 7 Performance plots (in %) of ML/DL models used in robotic systems for crop/weed discrimination task

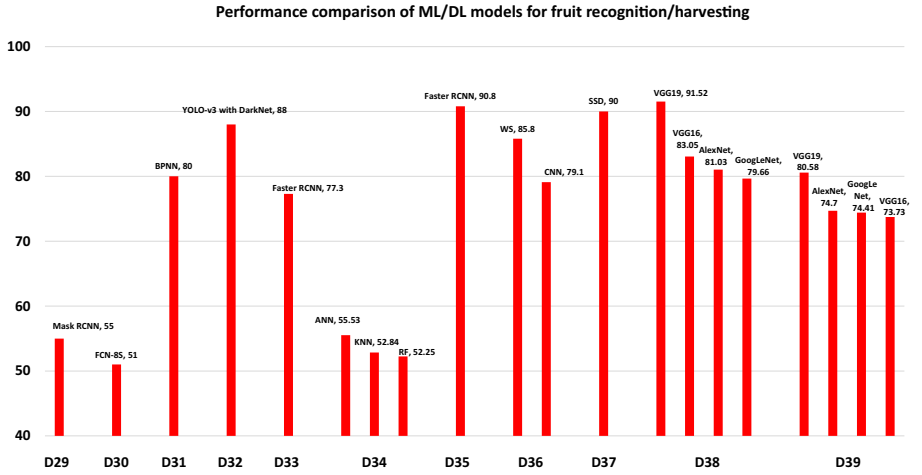


Fig. 8 Performance plots (in %) of ML/DL models used in robotic systems for fruit recognition and harvesting tasks

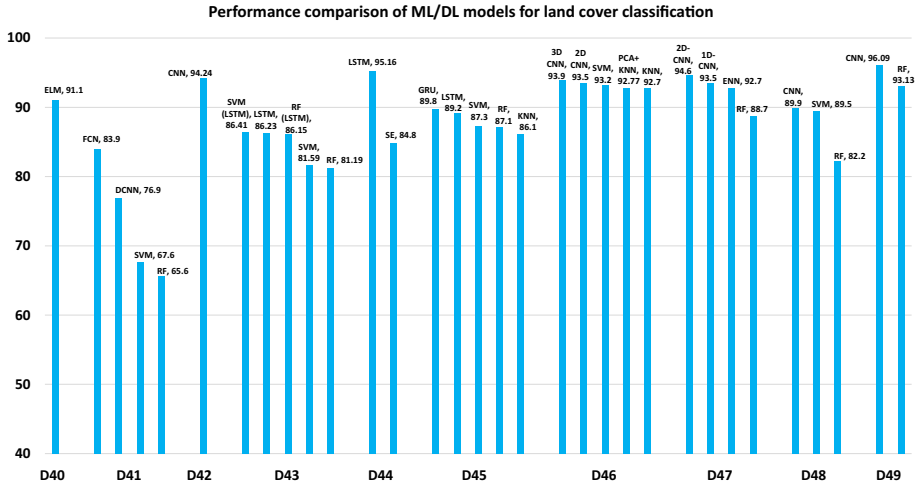


Fig. 9 Performance plots (in %) of ML/DL models used in robotic systems for the land cover classification

Appendix 2 Statements of Contribution DRC 16

The Appendix 2 includes the eight DRC 16 forms.

MASSEY UNIVERSITY
TE KUNENGA KI PŪREHUROA
UNIVERSITY OF NEW ZEALAND

GRADUATE RESEARCH SCHOOL

STATEMENT OF CONTRIBUTION
DOCTORATE WITH PUBLICATIONS/MANUSCRIPTS

We, the student and the student's main supervisor, certify that all co-authors have consented to their work being included in the thesis and they have accepted the student's contribution as indicated below in the Statement of Originality.

Student name:	Muhammad Hammad Saleem
Name and title of main supervisor:	Dr Khalid Arif
In which chapter is the manuscript/published work?	Chapter 2
What percentage of the manuscript/published work was contributed by the student?	80%

Describe the contribution that the student has made to the manuscript/published work:

The candidate performed the literature review and gathered the data. He also prepared the first draft for submission to journal based on suggestions from the co-author supervisors.

Please select one of the following three options:

- (●) The manuscript/published work is published or in press

  Please provide the full reference of the research output:

  M. H. Saleem, J. Potgieter, and K. M. Arif, "Plant disease detection and classification by deep learning," Plants, vol. 8, no. 11, p. 468, 2019.

- ( ) The manuscript is currently under review for publication

  Please provide the name of the journal:

- ( ) It is intended that the manuscript will be published, but it has not yet been submitted to a journal

Student's signature:	Muhammad Hammad Saleem — Digitally signed by Muhammad Hammad Saleem Date: 2022.07.29 18:18:52 +12'00'	Main supervisor's signature:	Khalid Arif — Digitally signed by Khalid Arif Date: 2022.07.29 22:39:50 +12'00'

This form should appear at the end of each thesis chapter/section/appendix submitted as a manuscript/ publication or collected as an appendix at the end of the thesis.

MASSEY UNIVERSITY
TE KUNENGA KI PŪREHUROA
UNIVERSITY OF NEW ZEALAND

GRADUATE RESEARCH SCHOOL

STATEMENT OF CONTRIBUTION
DOCTORATE WITH PUBLICATIONS/MANUSCRIPTS

We, the student and the student's main supervisor, certify that all co-authors have consented to their work being included in the thesis and they have accepted the student's contribution as indicated below in the Statement of Originality.

Student name:	Muhammad Hammad Saleem
Name and title of main supervisor:	Dr Khalid Arif
In which chapter is the manuscript/published work?	Chapter 3
What percentage of the manuscript/published work was contributed by the student?	80%

Describe the contribution that the student has made to the manuscript/published work:

The candidate designed the research and performed the experiments. He also prepared the first draft for submission to journal based on suggestions from the co-author supervisors.

Please select one of the following three options:

- (●) The manuscript/published work is published or in press

  Please provide the full reference of the research output:

  M. H. Saleem, J. Potgieter, and K. M. Arif, "Plant disease classification: A comparative evaluation of convolutional neural networks and deep learning optimizers," Plants, vol. 9, no. 10, p. 1319, 2020.

- ( ) The manuscript is currently under review for publication

  Please provide the name of the journal:

- ( ) It is intended that the manuscript will be published, but it has not yet been submitted to a journal

Student's signature:	Muhammad Hammad Saleem — Digitally signed by Muhammad Hammad Saleem Date: 2022.07.29 18:21:31 +12'00'	Main supervisor's signature:	Khalid Arif — Digitally signed by Khalid Arif Date: 2022.07.29 22:39:34 +12'00'

This form should appear at the end of each thesis chapter/section/appendix submitted as a manuscript/ publication or collected as an appendix at the end of the thesis.

MASSEY UNIVERSITY
TE KUNENGA KI PŪREHUROA
UNIVERSITY OF NEW ZEALAND

GRADUATE RESEARCH SCHOOL

STATEMENT OF CONTRIBUTION
DOCTORATE WITH PUBLICATIONS/MANUSCRIPTS

We, the student and the student's main supervisor, certify that all co-authors have consented to their work being included in the thesis and they have accepted the student's contribution as indicated below in the Statement of Originality.

Student name:	Muhammad Hammad Saleem
Name and title of main supervisor:	Dr Khalid Arif
In which chapter is the manuscript/published work?	Chapter 4
What percentage of the manuscript/published work was contributed by the student?	70%

Describe the contribution that the student has made to the manuscript/published work:

The candidate designed the research and performed the experiments with assistance from second author. He also prepared the first draft for submission to journal based on suggestions from the co-author supervisors.

Please select one of the following three options:

- (●) The manuscript/published work is published or in press

  Please provide the full reference of the research output:

  M. H. Saleem, S. Khanchi, J. Potgieter, and K. M. Arif, "Image-based plant disease identification by deep learning meta-architectures," Plants, vol. 9, no. 11, p. 1451, 2020.

- ( ) The manuscript is currently under review for publication

  Please provide the name of the journal:

- ( ) It is intended that the manuscript will be published, but it has not yet been submitted to a journal

Student's signature:	Muhammad Hammad Saleem — Digitally signed by Muhammad Hammad Saleem Date: 2022.07.29 18:23:32 +12'00'	Main supervisor's signature:	Khalid Arif — Digitally signed by Khalid Arif Date: 2022.07.29 22:40:05 +12'00'

This form should appear at the end of each thesis chapter/section/appendix submitted as a manuscript/ publication or collected as an appendix at the end of the thesis.

MASSEY UNIVERSITY
TE KUNENGA KI PŪREHUROA
UNIVERSITY OF NEW ZEALAND

GRADUATE RESEARCH SCHOOL

STATEMENT OF CONTRIBUTION
DOCTORATE WITH PUBLICATIONS/MANUSCRIPTS

We, the student and the student's main supervisor, certify that all co-authors have consented to their work being included in the thesis and they have accepted the student's contribution as indicated below in the Statement of Originality.

Student name:	Muhammad Hammad Saleem
Name and title of main supervisor:	Dr Khalid Arif
In which chapter is the manuscript/published work?	Chapter 5
What percentage of the manuscript/published work was contributed by the student?	70%

Describe the contribution that the student has made to the manuscript/published work:

The candidate designed this research and performed the experiments with assistance from second author. He also prepared the first draft for submission to journal based on suggestions from the co-author supervisors.

Please select one of the following three options:

- (●) The manuscript/published work is published or in press

  Please provide the full reference of the research output:

  M. H. Saleem, K. K. Velayudhan, J. Potgieter, and K. M. Arif, "Weed Identification by Single-Stage and Two-Stage Neural Networks: A Study on the Impact of Image Resizers and Weights Optimization Algorithms," Frontiers in Plant Science, vol. 13, 2022.

- ( ) The manuscript is currently under review for publication

  Please provide the name of the journal:

- ( ) It is intended that the manuscript will be published, but it has not yet been submitted to a journal

Student's signature:	Muhammad Hammad Saleem — Digitally signed by Muhammad Hammad Saleem Date: 2022.07.29 18:24:39 +12'00'	Main supervisor's signature:	Khalid Arif — Digitally signed by Khalid Arif Date: 2022.07.29 22:40:34 +12'00'

This form should appear at the end of each thesis chapter/section/appendix submitted as a manuscript/ publication or collected as an appendix at the end of the thesis.

MASSEY UNIVERSITY
TE KUNENGA KI PŪREHUROA
UNIVERSITY OF NEW ZEALAND

GRADUATE RESEARCH SCHOOL

STATEMENT OF CONTRIBUTION
DOCTORATE WITH PUBLICATIONS/MANUSCRIPTS

We, the student and the student's main supervisor, certify that all co-authors have consented to their work being included in the thesis and they have accepted the student's contribution as indicated below in the Statement of Originality.

Student name:	Muhammad Hammad Saleem
Name and title of main supervisor:	Dr Khalid Arif
In which chapter is the manuscript/published work?	Chapter 6
What percentage of the manuscript/published work was contributed by the student?	80%

Describe the contribution that the student has made to the manuscript/published work:

The candidate designed the research and performed the experiments. He also prepared the first draft for submission to journal based on suggestions from the co-author supervisors.

Please select one of the following three options:

- (●) The manuscript/published work is published or in press

  Please provide the full reference of the research output:

  M. H. Saleem, J. Potgieter, and K. M. Arif, "Weed Detection by Faster RCNN Model: An Enhanced Anchor Box Approach," Agronomy, vol. 12, no. 7, p. 1580, 2022.

- ( ) The manuscript is currently under review for publication

  Please provide the name of the journal:

- ( ) It is intended that the manuscript will be published, but it has not yet been submitted to a journal

Student's signature:	Muhammad Hammad Saleem — Digitally signed by Muhammad Hammad Saleem Date: 2022.07.29 18:30:20 +12'00'	Main supervisor's signature:	Khalid Arif — Digitally signed by Khalid Arif Date: 2022.07.29 22:40:20 +12'00'

This form should appear at the end of each thesis chapter/section/appendix submitted as a manuscript/ publication or collected as an appendix at the end of the thesis.

MASSEY UNIVERSITY
TE KUNENGA KI PŪREHUROA
UNIVERSITY OF NEW ZEALAND

GRADUATE RESEARCH SCHOOL

STATEMENT OF CONTRIBUTION
DOCTORATE WITH PUBLICATIONS/MANUSCRIPTS

We, the student and the student's main supervisor, certify that all co-authors have consented to their work being included in the thesis and they have accepted the student's contribution as indicated below in the Statement of Originality.

Student name:	Muhammad Hammad Saleem
Name and title of main supervisor:	Dr Khalid Arif
In which chapter is the manuscript/published work?	Chapter 7
What percentage of the manuscript/published work was contributed by the student?	80%

Describe the contribution that the student has made to the manuscript/published work:

The candidate generated the dataset, designed the research and performed the experiments. He also prepared the first draft for submission to journal based on suggestions from the co-author supervisors.

Please select one of the following three options:

○	The manuscript/published work is published or in press Please provide the full reference of the research output:
◉	The manuscript is currently under review for publication Please provide the name of the journal: IEEE Access
○	It is intended that the manuscript will be published, but it has not yet been submitted to a journal

Student's signature:	Muhammad Hammad Saleem Digitally signed by Muhammad Hammad Saleem Date: 2022.07.29 18:31:00 +12'00'	Main supervisor's signature:	Khalid Arif Digitally signed by Khalid Arif Date: 2022.07.29 22:40:50 +12'00'

This form should appear at the end of each thesis chapter/section/appendix submitted as a manuscript/ publication or collected as an appendix at the end of the thesis.

MASSEY UNIVERSITY
TE KUNENGA KI PŪREHUROA
UNIVERSITY OF NEW ZEALAND

GRADUATE RESEARCH SCHOOL

STATEMENT OF CONTRIBUTION
DOCTORATE WITH PUBLICATIONS/MANUSCRIPTS

We, the student and the student's main supervisor, certify that all co-authors have consented to their work being included in the thesis and they have accepted the student's contribution as indicated below in the Statement of Originality.

Student name:	Muhammad Hammad Saleem
Name and title of main supervisor:	Dr Khalid Arif
In which chapter is the manuscript/published work?	Chapter 8
What percentage of the manuscript/published work was contributed by the student?	80%

Describe the contribution that the student has made to the manuscript/published work:

The candidate generated the dataset, designed the research and performed the experiments. He also prepared the first draft for submission to journal based on suggestions from the co-author supervisors.

Please select one of the following three options:

- ○ The manuscript/published work is published or in press
  Please provide the full reference of the research output:
- ◉ The manuscript is currently under review for publication
  Please provide the name of the journal:
  Frontiers in Plant Science
- ○ It is intended that the manuscript will be published, but it has not yet been submitted to a journal

Student's signature:	Muhammad Hammad Saleem Digitally signed by Muhammad Hammad Saleem Date: 2022.07.29 18:31:30 +12'00'	Main supervisor's signature:	Khalid Arif Digitally signed by Khalid Arif Date: 2022.07.29 22:38:55 +12'00'

This form should appear at the end of each thesis chapter/section/appendix submitted as a manuscript/ publication or collected as an appendix at the end of the thesis.

MASSEY UNIVERSITY
TE KUNENGA KI PŪREHUROA
UNIVERSITY OF NEW ZEALAND

GRADUATE RESEARCH SCHOOL

STATEMENT OF CONTRIBUTION
DOCTORATE WITH PUBLICATIONS/MANUSCRIPTS

We, the student and the student's main supervisor, certify that all co-authors have consented to their work being included in the thesis and they have accepted the student's contribution as indicated below in the Statement of Originality.

Student name:	Muhammad Hammad Saleem
Name and title of main supervisor:	Dr Khalid Arif
In which chapter is the manuscript/published work?	Appendix 1
What percentage of the manuscript/published work was contributed by the student?	80%

Describe the contribution that the student has made to the manuscript/published work:

The candidate performed the literature review and gathered the data. He also prepared the first draft for submission to journal based on suggestions from the co-author supervisors.

Please select one of the following three options:

- (●) The manuscript/published work is published or in press

  Please provide the full reference of the research output:

  M. H. Saleem, J. Potgieter, and K. M. Arif, "Automation in agriculture by machine and deep learning techniques: A review of recent developments," Precision Agriculture, vol. 22, no. 6, pp. 2053-2091, 2021.

- ( ) The manuscript is currently under review for publication

  Please provide the name of the journal:

- ( ) It is intended that the manuscript will be published, but it has not yet been submitted to a journal

Student's signature:	Muhammad Hammad Saleem Digitally signed by Muhammad Hammad Saleem Date: 2022.07.29 18:32:05 +12'00'	Main supervisor's signature:	Khalid Arif Digitally signed by Khalid Arif Date: 2022.07.29 22:39:15 +12'00'

This form should appear at the end of each thesis chapter/section/appendix submitted as a manuscript/ publication or collected as an appendix at the end of the thesis.

Appendix 3 Reprints and permissions

The Appendix 3 includes the permission of reprinting the article published in a journal named Precision Agriculture.

Appendix 3

SPRINGER NATURE LICENSE
TERMS AND CONDITIONS

Jul 26, 2022

This Agreement between Massey University -- Muhammad Hammad Saleem ("You") and Springer Nature ("Springer Nature") consists of your license details and the terms and conditions provided by Springer Nature and Copyright Clearance Center.

License Number	5356410769411
License date	Jul 26, 2022
Licensed Content Publisher	Springer Nature
Licensed Content Publication	Precision Agriculture
Licensed Content Title	Automation in Agriculture by Machine and Deep Learning Techniques: A Review of Recent Developments
Licensed Content Author	Muhammad Hammad Saleem et al
Licensed Content Date	Apr 21, 2021
Type of Use	Thesis/Dissertation
Requestor type	academic/university or research institute
Format	print and electronic
Portion	full article/chapter
Will you be translating?	no

Appendix 3

Circulation/distribution	500 - 999
Author of this Springer Nature content	yes
Title	PhD Student
Institution name	Massey University
Expected presentation date	Jul 2022
Requestor Location	Massey University 8A Alexander Avenue, Whakatāne, 3120 Whakatāne, 3120 New Zealand Attn: Massey University
Total	0.00 USD

Terms and Conditions

Springer Nature Customer Service Centre GmbH
Terms and Conditions

This agreement sets out the terms and conditions of the licence (the **Licence**) between you and **Springer Nature Customer Service Centre GmbH** (the **Licensor**). By clicking 'accept' and completing the transaction for the material (**Licensed Material**), you also confirm your acceptance of these terms and conditions.

1. Grant of License

1. 1. The Licensor grants you a personal, non-exclusive, non-transferable, world-wide licence to reproduce the Licensed Material for the purpose specified in your order only. Licences are granted for the specific use requested in the order and for no other use, subject to the conditions below.

1. 2. The Licensor warrants that it has, to the best of its knowledge, the rights to license reuse of the Licensed Material. However, you should ensure that the material you are requesting is original to the Licensor and does not carry the copyright of another entity (as credited in the published version).

1. 3. If the credit line on any part of the material you have requested indicates that it was reprinted or adapted with permission from another source, then you should also seek permission from that source to reuse the material.

2. Scope of Licence

2. 1. You may only use the Licensed Content in the manner and to the extent permitted by these Ts&Cs and any applicable laws.

2. 2. A separate licence may be required for any additional use of the Licensed Material, e.g. where a licence has been purchased for print only use, separate permission must be obtained for electronic re-use. Similarly, a licence is only valid in the language selected and does not apply for editions in other languages unless additional translation rights have been granted separately in the licence. Any content owned by third parties are expressly excluded from the licence.

2. 3. Similarly, rights for additional components such as custom editions and derivatives require additional permission and may be subject to an additional fee. Please apply to Journalpermissions@springernature.com/bookpermissions@springernature.com for these rights.

2. 4. Where permission has been granted **free of charge** for material in print, permission may also be granted for any electronic version of that work, provided that the material is incidental to your work as a whole and that the electronic version is essentially equivalent to, or substitutes for, the print version.

2. 5. An alternative scope of licence may apply to signatories of the STM Permissions Guidelines, as amended from time to time.

3. Duration of Licence

3. 1. A licence for is valid from the date of purchase ('Licence Date') at the end of the relevant period in the below table:

Scope of Licence	Duration of Licence
Post on a website	12 months
Presentations	12 months
Books and journals	Lifetime of the edition in the language purchased

4. Acknowledgement

4. 1. The Licensor's permission must be acknowledged next to the Licenced Material in print. In electronic form, this acknowledgement must be visible at the same time as the figures/tables/illustrations or abstract, and must be hyperlinked to the journal/book's homepage. Our required acknowledgement format is in the Appendix below.

5. Restrictions on use

5. 1. Use of the Licensed Material may be permitted for incidental promotional use and minor editing privileges e.g. minor adaptations of single figures, changes of format, colour and/or style where the adaptation is credited as set out in Appendix 1 below. Any other changes including but not limited to, cropping, adapting, omitting material that affect the meaning, intention or moral rights of the author are strictly prohibited.

5. 2. You must not use any Licensed Material as part of any design or trademark.

5. 3. Licensed Material may be used in Open Access Publications (OAP) before publication by Springer Nature, but any Licensed Material must be removed from OAP sites prior to final publication.

6. Ownership of Rights

6. 1. Licensed Material remains the property of either Licensor or the relevant third party and any rights not explicitly granted herein are expressly reserved.

7. Warranty

IN NO EVENT SHALL LICENSOR BE LIABLE TO YOU OR ANY OTHER PARTY OR ANY OTHER PERSON OR FOR ANY SPECIAL, CONSEQUENTIAL, INCIDENTAL OR INDIRECT DAMAGES, HOWEVER CAUSED, ARISING OUT OF OR IN CONNECTION WITH THE DOWNLOADING, VIEWING OR USE OF THE MATERIALS REGARDLESS OF THE FORM OF ACTION, WHETHER FOR BREACH OF CONTRACT, BREACH OF WARRANTY, TORT, NEGLIGENCE, INFRINGEMENT OR OTHERWISE (INCLUDING, WITHOUT LIMITATION, DAMAGES BASED ON LOSS OF PROFITS, DATA, FILES, USE, BUSINESS OPPORTUNITY OR CLAIMS OF THIRD PARTIES), AND
WHETHER OR NOT THE PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION SHALL APPLY NOTWITHSTANDING ANY FAILURE OF ESSENTIAL PURPOSE OF ANY LIMITED REMEDY PROVIDED HEREIN.

8. Limitations

8. 1. ***BOOKS ONLY:*** Where **'reuse in a dissertation/thesis'** has been selected the following terms apply: Print rights of the final author's accepted manuscript (for clarity, NOT the published version) for up to 100 copies, electronic rights for use only on a personal website or institutional repository as defined by the Sherpa guideline (www.sherpa.ac.uk/romeo/).

8. 2. For content reuse requests that qualify for permission under the STM Permissions Guidelines, which may be updated from time to time, the STM Permissions Guidelines supersede the terms and conditions contained in this licence.

9. Termination and Cancellation

9. 1. Licences will expire after the period shown in Clause 3 (above).

9. 2. Licensee reserves the right to terminate the Licence in the event that payment is not received in full or if there has been a breach of this agreement by you.

Appendix 1 — Acknowledgements:

For Journal Content:
Reprinted by permission from [**the Licensor**]: [**Journal Publisher** (e.g. Nature/Springer/Palgrave)] [**JOURNAL NAME**] [**REFERENCE CITATION** (Article name, Author(s) Name), [**COPYRIGHT**] (year of publication)

For **Advance Online Publication papers:**
Reprinted by permission from [**the Licensor**]: [**Journal Publisher** (e.g. Nature/Springer/Palgrave)] [**JOURNAL NAME**] [**REFERENCE CITATION** (Article name, Author(s) Name), [**COPYRIGHT**] (year of publication), advance online publication, day month year (doi: 10.1038/sj.[JOURNAL ACRONYM].)

For Adaptations/Translations:
Adapted/Translated by permission from [**the Licensor**]: [**Journal Publisher** (e.g. Nature/Springer/Palgrave)] [**JOURNAL NAME**] [**REFERENCE CITATION** (Article name, Author(s) Name), [**COPYRIGHT**] (year of publication)

Note: For any republication from the British Journal of Cancer, the following credit line style applies:

Reprinted/adapted/translated by permission from [**the Licensor**]: on behalf of Cancer Research UK: : [**Journal Publisher** (e.g. Nature/Springer/Palgrave)] [**JOURNAL NAME**] [**REFERENCE CITATION** (Article name, Author(s) Name), [**COPYRIGHT]** (year of publication)

For **Advance Online Publication** papers:
Reprinted by permission from The [**the Licensor**]: on behalf of Cancer Research UK: [**Journal Publisher** (e.g. Nature/Springer/Palgrave)] [**JOURNAL NAME**] [**REFERENCE CITATION** (Article name, Author(s) Name), [**COPYRIGHT**] (year of publication), advance online publication, day month year (doi: 10.1038/sj.[JOURNAL ACRONYM])

For Book content:
Reprinted/adapted by permission from [**the Licensor**]: [**Book Publisher** (e.g. Palgrave Macmillan, Springer etc) [**Book Title**] by [**Book author**(s)] [**COPYRIGHT**] (year of publication)

Other Conditions:

Version 1.3

Questions? customercare@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777.