Learning-based robotic manipulation for dynamic object handling

A thesis presented in partial fulfilment of the requirements for the degree of

Doctor of philosophy
in
mechatronic engineering

at the
School of Food and Advanced Technology,
Massey University, Turitea Campus,
Palmerston North,
New Zealand

Jacques J.P. Janse van Vuuren

September 2020

Abstract

Recent trends have shown that the lifecycles and production volumes of modern products are shortening. Consequently, many manufacturers subject to frequent change prefer flexible and reconfigurable production systems. Such schemes are often achieved by means of manual assembly, as conventional automated systems are perceived as lacking flexibility. Production lines that incorporate human workers are particularly common within consumer electronics and small appliances. Artificial intelligence (AI) is a possible avenue to achieve smart robotic automation in this context. In this research it is argued that a robust, autonomous object handling process plays a crucial role in future manufacturing systems that incorporate robotics—key to further closing the gap between manual and fully automated production.

Novel object grasping is a difficult task, confounded by many factors including object geometry, weight distribution, friction coefficients and deformation characteristics. Sensing and actuation accuracy can also significantly impact manipulation quality. Another challenge is understanding the relationship between these factors, a specific grasping strategy, the robotic arm and the employed end-effector. Manipulation has been a central research topic within robotics for many years. Some works focus on design, i.e. specifying a gripper-object interface such that the effects of imprecise gripper placement and other confounding control-related factors are mitigated. Many universal robotic gripper designs have been considered, including 3-fingered gripper designs, anthropomorphic grippers, granular jamming end-effectors and underactuated mechanisms. While such approaches have maintained some interest, contemporary works predominantly utilise machine learning in conjunction with imaging technologies and generic force-closure end-effectors. Neural networks that utilise supervised and unsupervised learning schemes with an RGB or RGB-D input make up the bulk of publications within this field. Though many solutions have been studied, automatically generating a robust grasp configuration for objects not known *a priori*, remains an open-ended problem. An element of this issue relates to a lack of objective performance metrics to quantify the effectiveness of a solution—which has traditionally driven the direction of community focus by highlighting gaps in the state-of-the-art.

This research employs monocular vision and deep learning to generate—and select from—a set of hypothesis grasps. A significant portion of this research relates to the process by which a final grasp is selected. Grasp synthesis is achieved by sampling the workspace using convolutional neural networks trained to recognise prospective grasp areas. Each potential pose is evaluated by the proposed method in conjunction with other input modalities—such as load-cells and an alternate perspective. To overcome human bias and build upon traditional metrics, scores are established to objectively quantify the quality of an executed grasp trial. Learning frameworks that aim to maximise for these scores are employed in the selection process to improve performance. The proposed methodology and associated metrics are empirically evaluated.

A physical prototype system was constructed, employing a Dobot Magician robotic manipulator, vision enclosure, imaging system, conveyor, sensing unit and control system. Over 4,000 trials were conducted utilising 100 objects. Experimentation showed that robotic manipulation quality could be improved by 10.3% when selecting to optimise for the proposed metrics—quantified by a metric related to translational error. Trials further demonstrated a grasp success rate of 99.3% for known objects and 98.9% for objects for

which *a priori* information is unavailable. For unknown objects, this equated to an improvement of approximately 10% relative to other similar methodologies in literature. A 5.3% reduction in grasp rate was observed when removing the metrics as selection criteria for the prototype system. The system operated at approximately 1 Hz when contemporary hardware was employed. Experimentation demonstrated that selecting a grasp pose based on the proposed metrics improved grasp rates by up to 4.6% for known objects and 2.5% for unknown objects—compared to selecting for grasp rate alone.

This project was sponsored by the Richard and Mary Earle Technology Trust, the Ken and Elizabeth Powell Bursary and the Massey University Foundation. Without the financial support provided by these entities, it would not have been possible to construct the physical robotic system used for testing and experimentation. This research adds to the field of robotic manipulation, contributing to topics on *grasp-induced error analysis*, *post-grasp error minimisation*, *grasp synthesis framework design* and *general grasp synthesis*. Three journal publications and one IEEE Xplore paper have been published as a result of this research.

Acknowledgements

Firstly, I would like to express my gratitude toward my main supervisor, Dr. Liqiong Tang, for her patience, trenchant yet constructive criticism and guidance throughout. In comparing the standard of this thesis to my previous publications, it is clear that her influence has immensely improved my writing and general academic ability. I appreciate the many hours she has dedicated toward my professional development and it has been a pleasure to work with her. Our visit to China was particularly memorable and helped to shape my understanding of academia and the scientific method. I am also appreciative of my co-supervisors, Assoc. Prof. Ibrahim Al-Bahadly and Dr. Khalid Mahmood Arif. Their feedback strongly influenced the direction of this work in the early stages.

I extend my thanks to the School of Food and Advanced Technology, Massey University for providing the resources and infrastructure for this research. The contribution of Dr. Morio Fukuoka is also noted. I enjoyed our occasional project and non-project related discussions.

Thank you to my wife, Katy Johnston. Her support has enabled me to reach this milestone. Her academic mentality facilitated many discussions related to this research and other subjects surrounding this research. Consequently, she has a much greater understanding of machine learning than other veterinarians. She has been the main source of income for our family over the past few years and I look forward to reciprocating while she continues with further study in the future.

Finally, I would like to thank my family and others involved in this project that I have not mentioned directly—they have helped make this research possible and contributed to its success.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Sorted alphabetically:

ACRV	The Australian Research Council Centre of Excellence for Robotic Vision
AI	Artificial intelligence
ANN	Artificial neural network
APC	Amazon Picking Challenge
CNN	Convolutional neural network
COG	Centre of gravity
CUDA	Compute unified device architecture
DIP	Digital image processing
EURON	European Robotics Research Network
FAST	Features from accelerated segment test
FPS	Frames per second
GFLOP	Giga floating point operations per second
GPU	Graphics processing unit
IFR	International Federation of Robotics
ILSVRC	ImageNet large scale visual recognition challenge
IoT	Internet of things
IoU	Intersection over union
ITIF	Information Technology & Innovation Foundation
KNN	K-nearest neighbour
kPAM	Keypoint affordance for category-level robotic manipulation
ML	Machine learning
MLR	Multiple linear regression
MV	Machine vision
NZ	New Zealand
PLC	Programmable logic controller
RMSE	Root mean squared error
SIFT	Scale invariant feature transform
SURF	Speeded up robust features
SVM	Support vector machine
YCB	Yale-CMU-Berkeley
YOLO	You only look once

Chapter 1

Introduction

1.1 General introduction

Autonomous object manipulation is a central task within the field of robotics. Study in this area has traditionally been focused on domestic applications, such as dishwasher unloading [1], cloth manipulation [2], bed-making [3], automated cooking [4, 5], service robotics [6-9], clutter clearing [10, 11] and general household-related grasping [12-14]. Industry-oriented manipulation has gained significant popularity over the past decade, partly because of interest from industry giants—namely Amazon and Google. The Amazon Picking Challenge (APC) [15] for example, posed a pick-and-stow task within unstructured, cluttered environments, centred around warehouse automation. This challenge has prompted many works to integrate object recognition, object pose estimation, grasp and motion planning and error detection and recovery [16-22]. Some other examples where automatic grasping has been studied for industrial purposes are shown in Figure 1.1 and include garbage sorting [23], a checkout robot application [24], mineral sorting [25], autonomous micromanipulation [26], under water object manipulation [27], assembly and kitting tasks [28], mobile manipulator applications [29, 30], library automation [31] and general grasping for industry [32-35].

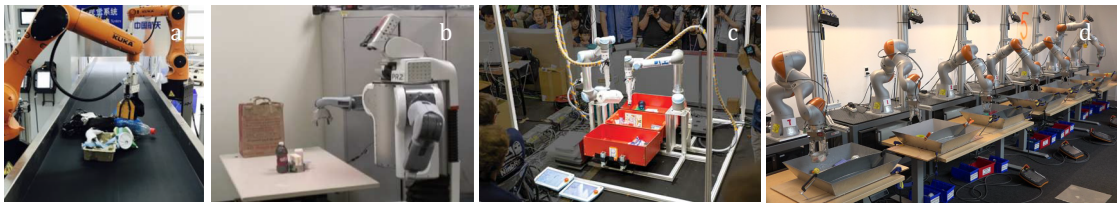


***Figure 1.1*. (a)—garbage sorting robot. Source: Zhihong, Hebin, Yanbo, Binyan and Yu [23]. (b)—checkout robot. Source: Klingbeil *et al.* [24]. (c)—dual-arm robotic picking system developed for the 2017 Amazon Robotics Challenge. Source: Schwarz *et al.* [17]. (d)—large-scale data collection framework for Google/Berkeley project. Source: Kalashnikov *et al.* [36].**

Modern-day industrial robots can carry out many complex manipulation and assembly tasks through careful hand-scripted actions, ranging from pick-and-place applications [37, 38] to the assembly of low-voltage circuit breakers [39]. However, autonomously interacting with objects not known *a priori* without explicit instruction and in an intelligent manner, remains an open problem. Automating this task is extremely difficult, as a good grasp is related to object shape, size, weight, weight-distribution, surface properties, friction coefficients, object deformability, etc., and can be severely affected by perception and actuation accuracy and noise [40-43]. Acquiring such information in practice is not trivial either and may not be feasible for some applications. Moreover, the relationship between these variables and a specific grasping strategy, robotic hardware and gripper is not always clear.

Novel object grasping is a well-studied field. Earlier works focused on universal grippers, designed to reduce the importance of end-effector placement. Odhner, Ma and Dollar for example, proposed an underactuated, 2-fingered gripper design aimed at emulating the human strategy for pinching objects from a flat surface [44]. They showed good

performance for small, thin objects like keys or coins—which is a common issue for many contemporary works. Under-actuated gripper designs have been well-researched [45-47]. More complex grippers have also been proposed, including, 3-fingered designs [48, 49], 4-fingered designs [50-52] and anthropomorphic hands [53, 54]. Brown *et al.* considered a granular jamming end-effector design, in which a mass is pressed onto an object [55]. By eliminating the air within the mass via vacuum, the shape conforms to the candidate object. Their methodology showed excellent performance for a wide range of objects and significantly reduced hardware and software complexity, but lacked gripping force for round, flat and small objects. Complex designs tend to excel at very specific tasks to the detriment of generalised performance. As such, industrial grippers and grippers used within this field for research tend to be predominantly 2-fingered, parallel jaw end-effectors [24, 56-62]. This research also concentrates on 2-fingered, force-closure grasps.

Recently, well-known, and well-resourced institutes such as Google, MIT, NASA, NVIDIA and UC Berkeley have resourced novel object handling research. Shi and Koonjul investigated the use of an anthropomorphic hand-based grasp planner for a variety of general automotive assembly applications at NASA-JSC [28]. The Google affiliated work by Levine, Pastor, Krizhevsky and Quillen saw the implementation of an unprecedented, large-scale dataset generation framework [35]. By utilising 14 robotic manipulators they collected more than 800,000 grasp attempt samples over the course of two months, which allowed them to train sophisticated, self-supervised convolutional neural networks (CNN) for grasp success prediction in clutter. Similarly, Kalashnikov *et al.* leveraged over 580,000 real-world grasp samples to train a deep reinforcement learning framework to perform closed-loop grasping on previously unseen objects [36]. Associated with MIT, Manuelli, Gao, Florence and Tedrake proposed kPAM—a novel task-specific representation of objects that ignored task-irrelevant geometry [12]. By representing an object in terms of semantic keypoints, they achieved purposeful robotic manipulation of novel objects within a known category. Tremblay *et al.* provide an example of work supported by NVIDIA [14]. They trained a 6-DOF pose estimator for known objects from a single RGB webcam image, using a mix of 120,000 real-world and synthetic data.

In general, machine learning (ML)-based, grasp synthesis methodologies have become over-represented within object handling literature [63, 64]. ML has been used to predict grasp pose in various ways—pioneered by Saxena, Driemeyer, Kearns and Ng [59]. They predicted grasping locations directly from 3D sensor data using a supervised learning technique. Later, Jiang, Moseson and Saxena [65] approached 2-fingered grasp synthesis by representing a grasp in terms of an oriented rectangle directly related to gripper placement. A plethora of works have since made use of analogous representations. Pinto and Gupta for example, collected 50,000 datapoints in terms of a 2-dimensional rectangular representation [60]. Sun, Yu, Liu and Gu employed a similar representation using depth information [66]. Further work by Lenz, Lee and Saxena effectively reduced grasp synthesis into a 2-stage generate-and-test task using their earlier representation [13]. First, a shallow classifier was used to quickly generate candidate grasping windows. A deeper scoring network was then employed to discriminate between windows for grasp selection. Their dataset—the Cornell Grasp Detection Dataset [67]—has since been used prolifically [40, 58, 66, 68-73]. Other grasp representations also exist, namely 3D pose estimation [6, 8, 14, 18, 74], a 2-fingered line representation [75, 76], feature points [24, 35, 59, 77] and others [11, 16, 25, 27].

Non-ML methodologies have also been investigated, for instance, Schwarz *et al.* segmented objects using ML but selected grasps heuristically [17], Militaru, Mezei and Tamas matched candidate objects with their known 3D models [6], Harada *et al.* approximated objects using cylinders [56] and S. Anton and F. Anton derived grasps by approximating object topology [32]. Despite the success of many such works, ML has significantly redefined the state-of-the-art, both in terms of computational efficiency and generalised performance. ML often performs at real-time, compared to non-ML which can take much longer. Redmon and Angelova illustrate this with their single-pass regression CNN that predicts grasps directly from sensor data [69]. Their methodology performed at 13 frames per second (FPS) and achieved 88.0% classification accuracy on the Cornell Grasp Detection Dataset. Conversely, the non-ML system proposed by Arruda, Wyatt and Kopicki [78] was documented to take 1.2 seconds to compute a grasp once at least two images were taken of the object at different viewpoints. On average, their system required 4.92 viewpoints for grasp planning and trials suggested an 80.4% grasp success rate.

Throughout object manipulation literature, the effectiveness of a grasping system or methodology is measured in various ways. Many works cite the rate at which their system was trialled to successfully grasp objects, although the definition of a successful grasp tends to be author dependent. Some make use of force sensors [45, 54, 79, 80], while many consider a grasp attempt successful if the object can be lifted above some pre-defined height [1, 16, 42, 51, 52, 59, 60, 65, 72, 81]. Some works check post-grasp finger position [35], while others employ a shake protocol [61, 82]. Several works will grasp an object and consider the trial a success if it can be dropped on command, put in a bin or complete some task [9, 12, 24, 36, 53, 62]. Morales, Chinellato, Fagg and del Pobil [83] grade grasp outcome from A–E through classification based on estimated reliability. By and large, most works simply frame a grasp trial in terms of a $pass$ or $fail$ [10, 13, 41, 44, 48, 55, 70, 75, 76, 84, 85]—without explicitly defining what is meant by those terms. Although this binary metric can be useful for comparison and training purposes, it has not been shown to be well-correlated with object grasp quality, handling quality or placement quality. This is made evident when good results in terms of grasp rate are achieved, but an object is clearly displaced by the grasping process itself [1, 12, 35, 40, 48, 50-52, 59, 65]—to the detriment of manipulation and placement quality. Currently, state-of-the-art novel object grasp rates sit between 85-95%. Refer to Table 1.1 for more details.

***Table 1.1*. Cited grasping success rates of various 2-fingered gripper-based works, where objects were physically trialled in isolation. Number of trials per object, methodological approach and testing conditions varied.**

Year	Authors	Approach	Grasping success rate (%)	
			Known objects	Unknown objects
2007, 2008	Saxena *at al.* [1, 59]	2+ RGB. Logistic regression	90.0	87.8
2008	Saxena *et al.* [86]	RGBD. Probabilistic model	-	76.0
2010	Le *et al.* [87]	RGBD. Supervised learning	-	81.9
2011	Klingbeil *et al.* [24]	RGBD. Grasping function	-	91.6
2011	Jiang *et al.* [65]	RGBD. Supervised learning	-	87.9
2014	Kopicki *et al.* [81]	RGBD. Supervised learning	88.0	86.0
2015	Sun *et al.* [66]	RGBD. Supervised learning	86.0	86.0
2015	ten Pas *et al.* [62]	RGBD. Supervised learning	-	88.0
2016	Pinto *et al.* [61]	RGB. Adversarial learning	82.0	82.0
2016	Pinto *et al.* [60]	RGB. Unsupervised learning	73.0	66.0
2016	Johns *et al.* [42]	RGBD. Supervised learning	-	80.3
2017	Mahler *et al.* [40]	RGBD. Supervised learning	93.0	80.0

2017	Viereck *et al.* [75]	RGBD. Supervised learning	-	97.5
2018	Lévesque *et al.* [88]	RGBD. Scooping grasp	-	84.1
2018	Chu *et al.* [70]	RGBD. Supervised learning	-	89.0
2018	Kalashnikov *et al.* [36]	RGB. Reinforcement learning	-	96.0
2019	James *et al.* [89]	RGB. Reinforcement learning	-	91.0
2019	Manuelli *et al.* [12]	RGBD. Reinforcement learning	-	91.7 (mugs) 98.3 (shoes)

Automated grasp synthesis is typically framed as a search problem—find an optimal grasp from a potentially infinite number of viable grasps—often approached as a generate-and-test task [13, 41, 42, 65, 66, 68]. By sampling the workspace, an initial subset of meaningful hypothesis grasps can be generated. Each sample within this subset can then be scored to facilitate a final selection. This research aims to build on such works by improving the way in which in the grasp hypothesis subset is pruned. New metrics are defined that measure grasp quality and more comprehensively describe grasp outcome than the traditional $pass/fail$ classification. The proposed metrics are continuous, which is useful for network training. Moreover, such metrics quantify the effectiveness of a grasp synthesis methodology—providing a useful baseline for objective comparison.

This research studies a scalable, manipulator-agnostic methodology that facilitates both good grasping and good handling of unknown objects within an industrial context. In contrast to many systems that utilise depth information [6, 20] , 3D models of objects [81, 90] or wrist-mounted sensors [19, 78], the proposed system operates on raw, monocular observations of the scene. A 3-staged learning framework is proposed. Stage one locates and classifies objects within the workspace, identifying known objects and cataloguing unknown objects. Stage two generates potential grasp areas for the detected objects, in terms of orientation and pose. Stage three integrates various sources of information, which are used to rank potential grasp areas. The top-ranked grasp may then be passed to the robotic manipulator for implementation. To validate the proposed methodology, a prototype was constructed. From 4,000 trials, it was demonstrated quantitatively that the proposed methodology can grasp small objects not seen during training, 98.9% of the time. It is shown that training to optimise for the proposed metrics can improve $pass/fail$ grasping rates by 6.0% for known objects and 5.3% for unknown objects—compared to the baseline. In addition, selecting for the proposed metrics improved grasp rates by 2.7% compared to selecting for grasp rate itself. An analysis of the stage three dataset revealed a strong relationship between the proposed metrics and the traditional $pass/fail$ notion of grasp outcome. Qualitative testing revealed subtle grasping behaviours exhibited by the proposed methodology that improved grasp quality and resultant performance.

1.2 Motivation for grasp-induced error metrics

Currently, the field of robotic manipulation lacks standardisation—particularly in the form of shared benchmarks and performance metrics. This issue makes it difficult to assess, compare or reproduce experimental results, and has been a known issue throughout literature for some time [91-93]. Mahler *et al.* suggested this difficulty stems, at least partly, from variations in experimental protocol, assumptions and differences in physical hardware, e.g., sensors, lighting, robotic arms, grippers and objects [93].

Various object test sets have been proposed to address the lack of comparable results within manipulation research. Such sets aim to provide a baseline for comparison by trialling a

manipulation system on the same set of objects. The YCB object set for example, consists of 75 objects categorised by utility, e.g., food items, kitchen items, tool items, shape items and task items [94, 95]. Leitner at al. proposed the ACRV picking benchmark comprised of 42 widely available objects, in addition to evaluation protocols and suggested arrangements [96, 97]. Although many such sets are commonly available, standardised use has not been adopted by the wider community. This has mainly been attributed to broad ranging methodological differences and hardware constraints that limit object applicability. Because of this, many researchers design their own object test set, often centred around common household or common laboratory objects [46, 54, 55, 82, 85, 98]. Generally, the object pool used for testing is well-documented. Simulation tools that build an environment for objective manipulation analysis have also been proposed, e.g., OpenGRASP [99], GraspIt! [100] and VisGraB [101].

Shared datasets have gained traction recently. The Cornell Grasp Detection Dataset for instance, has been used extensively for training and testing [40, 58, 66, 70-73]. Some works even assess their methodology via dataset performance alone [18, 20, 58, 68, 73]. In such cases, dataset accuracy is generally interpreted as a potential grasp rate—without conducting physical trials for validation. Redmon and Angelova illustrated this. They proposed a single-pass CNN that significantly improved grasp recognition performance in terms of computational efficiency and dataset performance—achieving 88% image-wise classification accuracy on the Cornell dataset [69]. Even though they extended the state-of-the-art on real recorded sensor data, their method was not physically validated. Dataset performance can provide an excellent platform for comparison, exemplified by the success of large-scale competition datasets for object recognition within the vision community, e.g., Pascal VOC [102, 103], ImageNet [104], COCO [105], etc. Unfortunately, applying the same approach to manipulation research relies on the assumption that dataset performance translates into real-world performance. Evidently this is not the case as many works report disparities between real-world grasp rates and dataset accuracy. Sun, Yu, Liu and Gu for instance, noted an 11% disparity between dataset classification accuracy and physical trial outcome [66]. Watson, Hughes and Lida reproduced the work of Redmon and Angelova using a Baxter research robot [106]. Their trials revealed a grasp rate of 62%—a 26% drop in expected performance. Commonly, datasets within this field are hand annotated [12, 20, 65, 68, 82, 107]. This is problematic because training for dataset performance tunes a methodology toward grasps the human creators consider optimal for real-world implementation. Moreover, quantifying the degree to which such annotated grasps map to a physical system is extremely difficult—as reflected by the various works that report similar performances with differing datasets. Due to the problems surrounding subjective datasets, some works have tried to avoid human bias altogether. Zeng *et al.* demonstrated that model-free, deep reinforcement learning can be used to learn complex grasping behaviours through trial and error [76]. Similarly, Pinto and Gupta employed self-supervision for over 700 robot hours [60]. Contemporary approaches that utilise hand-engineered labels still significantly outperform other methodologies.

To overcome human bias, this work proposes the institution of objective measures of grasp performance. By measuring the error introduced by a grasp trial itself, metrics can be established to assess grasp outcome, quantify grasp quality and provide useful criteria for network training. Traditionally, works throughout literature have focused on the notion of grasp success—where a physically attempted grasp is considered either a $pass$ or $fail$. Grasp-induced error metrics move away from this binary measure by providing a more comprehensive, continuous description of trial outcome. This is particularly useful for

generate-and-test methodologies in which the goal is to autonomously select a sample from the hypothesis grasp pool likely to result in the desired outcome. Furthermore, well-defined objectives for learning-based approaches can significantly impact performance. To help drive an algorithm toward an optimal solution, it is crucial to have measures that adequately describe the quality of a proposed solution.

1.3 Motivation for an industrially focused grasping methodology

The number of robotic arms utilised by industry has risen significantly over the past decade—particularly within the domains of spot welding, spray painting, electronic testing, metrology, assembly and machining [108]. The International Federation of Robotics (IFR) estimates the worldwide number of new robot arm installs in 2018 was over 400,000 units [109]. Compared to 2008, this represents a 370% increase in the number of installs. Moreover, the IFR predicts annual global installs to exceed 570,000 units by 2022. A more comprehensive view of installation trends is illustrated in Figure 1.2. The bulk of new installs in 2018 were related to automotive, electrical and electronic applications (Figure 1.3), although the largest growth was observed in food related installs, increasing by 32% between 2016 and 2018 [110]. There has also been an increase in the number of medical service robots installed, with 5,100 installs in 2018 compared to 7,200 in 2019. This number is forecasted to increase to 19,700 unit installs by 2020.

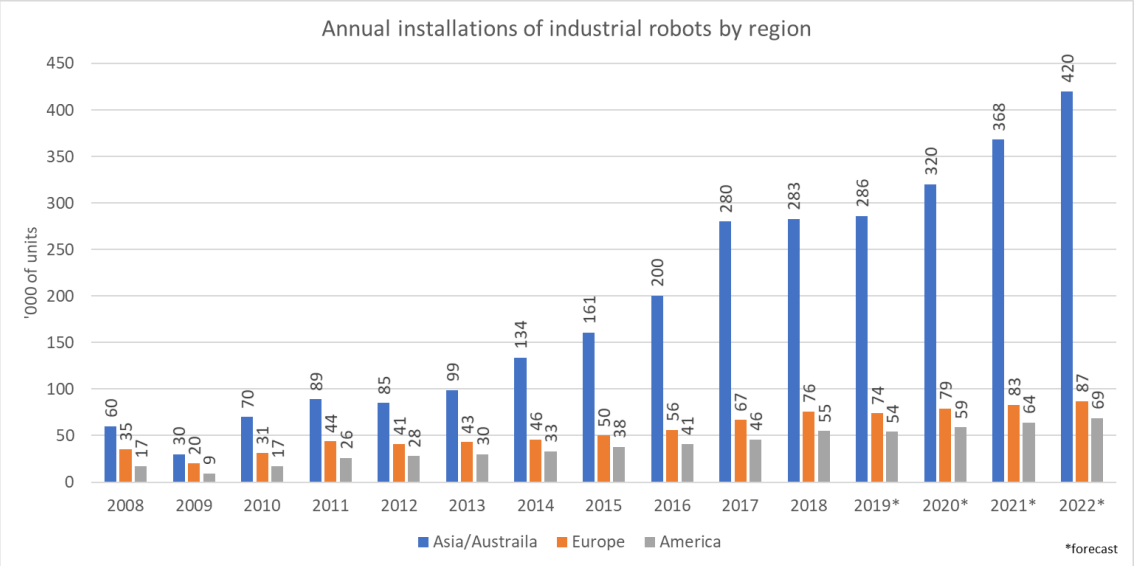


***Figure 1.2*. Estimated number of industrial robots installed per year by region. Source: International Federation of Robotics [109, 110].**

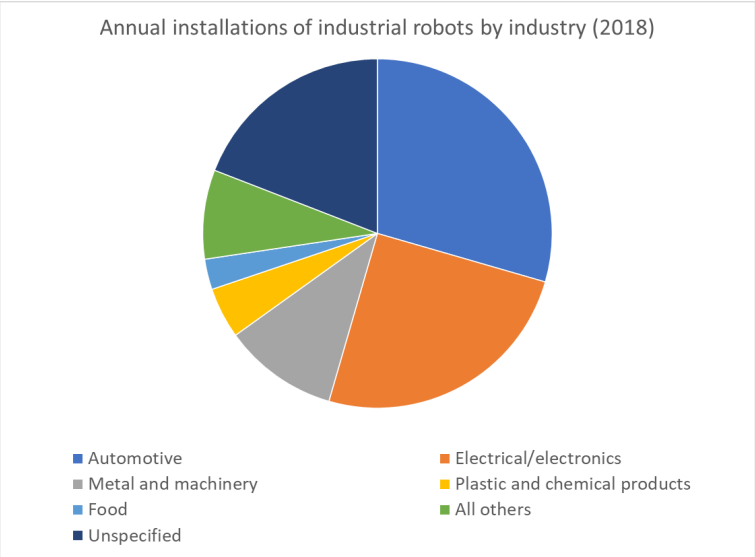


***Figure 1.3*. Estimated annual worldwide installations of industrial robots by industry, 2018. Source: International Federation of Robotics [109, 110].**

Modern factory automation is moving away from mass production due to the shortening of lifecycles and production volumes of many new products—influenced by consumers. This trend has been well-documented [108, 111-118]. For an in-depth review of current technologies and emerging trends within manufacture, the reader is referred to the overview provided by Dotoli, Fay, Miśkowiczc and Seatzu [119]. This shift, labelled Industry 4.0, introduces intelligent information and communication technologies with the goal of increasing the flexibility and scalability of production systems. McLean, Walker and Bright for example, proposed a disturbance mitigation approach for reconfigurable production systems based on artificial neural networks [120]. They demonstrated that artificial intelligence can be used to help manage perturbations such as shortage of material, breakdown of production elements and excessive tool wear. Ferrati, Nardi, Settimi, Marino and Pallottino proposed a smart planning algorithm that improved the autonomy of mobile robots in unstructured, multi-robot environments [30]. To help meet the demand for customisable production, many manufacturers are moving toward flexible and reconfigurable robotic systems. ABB for instance, have developed a dual-arm robotic concept designed to collaborate with human workers in agile production environments subject to regular change [121, 122]. Their concept was focused on small parts assembly within standard manual workstation arrangements. Generally, the flexibility of the control software of such robotic concepts is not considered to the same degree as the physical design.

To manage uncertainty surrounding product range, volume and lifespan, production systems prone to frequent change tend toward manual labour. As a result, manual production is common in low-wage countries. China currently represents an excellent example within this context. Considering that China is the world's largest trading and manufacture nation—accounting for 11.4% of global goods traded in 2017 [123]—the number of industrial robots per factory worker is relatively low. The Information Technology & Innovation Foundation (ITIF) estimated this figure at approximately 100 industrial robots per 10,000 workers in 2017 [124] (Figure 1.4). Korea had the largest number of industrial robots per worker in 2017, estimated at 710 units per 10,000 workers.

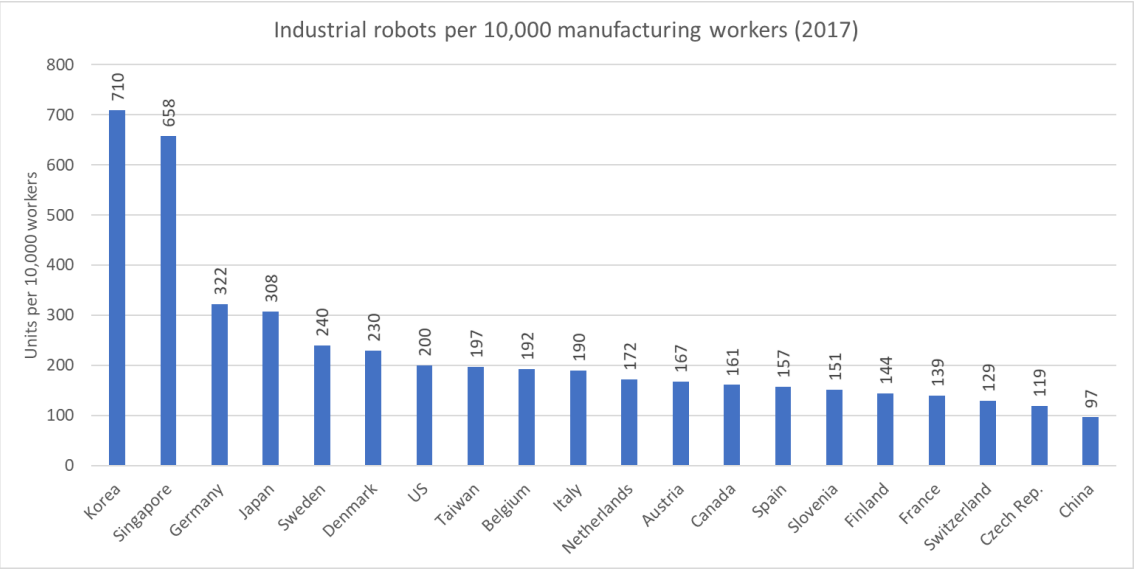


Figure 1.4. **Estimated number of industrial robots per 10,000 manufacturing workers by country, 2017. Source: Information Technology & Innovation Foundation (ITIF) [124].**

The number of industrial robots in China is rapidly growing. China alone installed over 166,000 new robots in 2018, which accounted for approximately 40% of all new robot installs worldwide [110]—more than four times the number installed in the United States. The number of industrial robots installed annually by market is illustrated in Figure 1.5. With such large volumes of new robot installs, the ratio between industrial robot and worker is rapidly increasing. In 2016 China surpassed the average global number of industrial robots per 10,000 workers—Figure 1.6. Moreover, the estimated operational stock of industrial robots in China grew exponentially between 2008 and 2018—Figure 1.7.

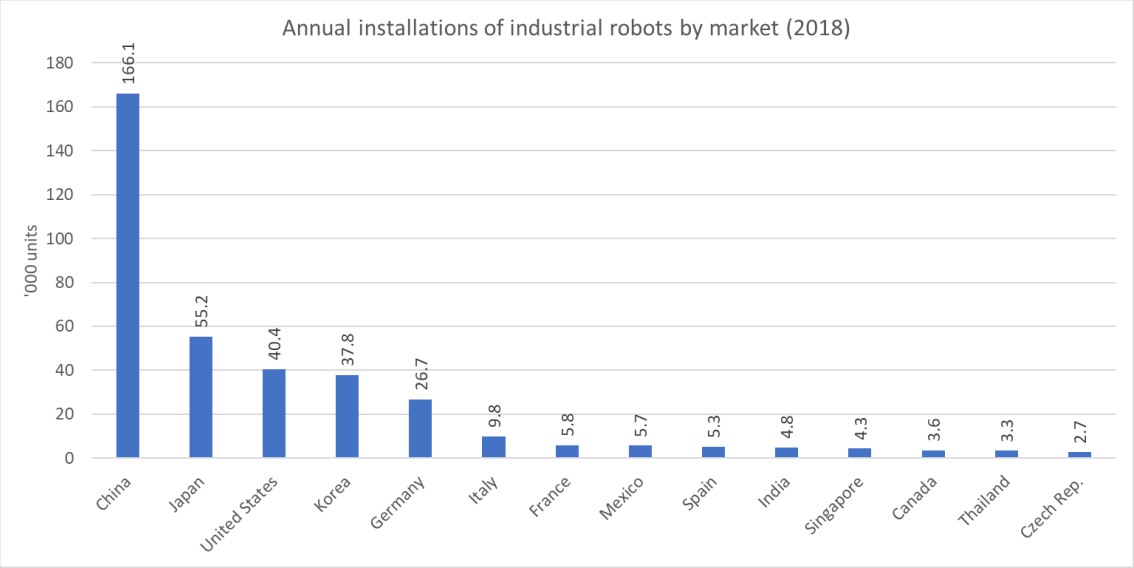


Figure 1.5. **Estimated number of industrial robots installed per year by market, 2018. Source: International Federation of Robotics [109, 110].**

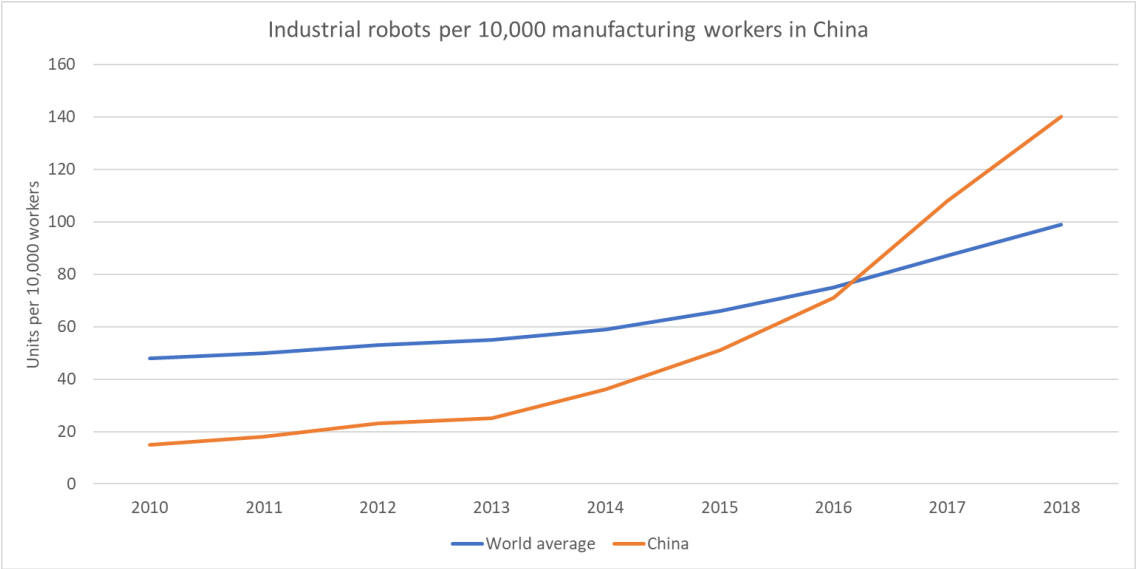


***Figure 1.6*. Estimated number of industrial robots per 10,000 manufacturing workers, by year. Source: International Federation of Robotics [109, 110].**

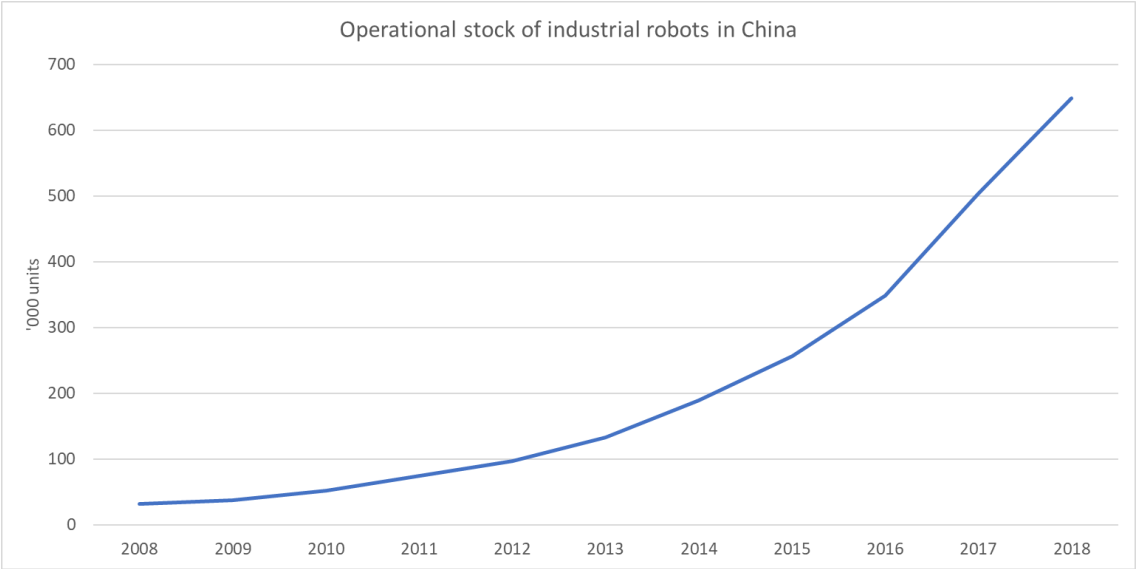


***Figure 1.7*. Estimated number of industrial robots operating in China by year. Source: International Federation of Robotics [109, 110].**

The applications of new robots installed in China often centre around handling tasks—which make up nearly a quarter of all application-specific installs, illustrated in Figure 1.8. While China shows the largest growth in terms of industrial robot automation, many other countries share a similar trend. India, for example, experienced a 39% increase in the number of units installed in 2018, compared to the previous year [125]. Even countries that already have a high level of robotic automation continue to see a rise in the number of annual installs. Japan, for instance, has had an average annual growth rate of 17% since 2013.

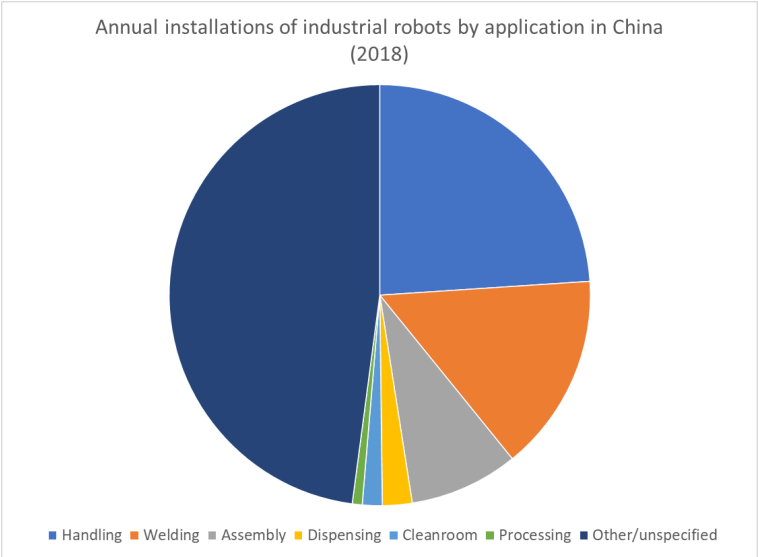


Figure 1.8. **Estimated annual installations of industrial robots by application in China, 2018. Source: International Federation of Robotics [109, 110].**

This research posits that a robust, autonomous handling system plays a crucial role in further closing the gap between manual and automated assembly for future production systems. Autonomous handling can greatly aid in a wide range of tasks, particularly for small-part assembly, e.g., the assembly of low-voltage circuit breakers [39], medical syringes [126], thermal relays [127], small automotive componentry [121], small electronic devices [128] and small consumer brushes [129]. Such a system may also be useful for product testing, nut driving, press fitting, product insertion, pick-and-place, material removal [130], part sorting and loading [131] and laboratory robot-scientist collaboration [132]. This research aims to investigate and develop a robust methodology for autonomous novel object handling that may be used in conjunction with other technologies to further respond to the high demand for automated manufacturing systems due to the reduced lifecycles of many modern products. This work is particularly focused on industrial application and production line integration, intended for the handling and assembly of small parts at this stage of development—although significant attention is also given to scalability and flexibility.

1.4 Problem formulation

The manufacture industry faces a continuing trend toward more complex products, an increase in product range, smaller batch sizes and reduced lifecycles [108, 111-118]. Previous studies reveal that multi-product production environments often achieve adequate flexibility, adaptability and reconfiguration potential through human workers; robotic systems are also utilised but they do require frequent manual attention. To effectively enable robotic automation within this context, reliable control theories and methodologies capable of accurately grasping and manipulating new objects are crucial. Such solutions are particularly relevant today, as the large-scale introduction of industrial robots is prevalent in many countries.

Autonomous handling is inherently a detection problem, provided an object has been correctly recognised. Once a desired object has been isolated within the workspace, a grasp location must be computed within a timely manner that will allow the object to be handled

reliably and precisely. Moreover, such a system must cater to a wide range of objects and promote compatibility.

The key challenges identified in this research are:

1. *Object identification*—difficulties in registering and precisely locating a present object within the region of interest.
2. *Object recognition*—correct categorisation of an object if it is known and appropriate classification of an object if it is novel.
3. *Grasp reliability*—the rate at which objects are grasped and handled successfully.
4. *Grasp quality*—the quality with which objects are grasped and handled.
5. *Scope of application*—difficulties in establishing a methodology that caters to wide ranging differences in object geometry, material, friction coefficients, etc.
6. *Processing time*—low processing time is crucial within an industrial setting to maintain high throughput.
7. *Scalability and integration*—issues surrounding the implementation of a handling system with existing hardware.

To evaluate the proposed methods and methodologies, the following criteria are set. A detailed discussion is presented in Chapter 10.

1. Empirical grasp rate of known objects.
2. Empirical grasp rate of unknown objects.
3. Neural network performance.
4. Grasp quality (quantitative/qualitative).
5. Computation time.
6. Scalability/ease of integration (qualitative).

1.5 Research scope and objectives

Robotics and AI cover many frontier challenges and span multiple disciplines. This research focuses primarily on the technical aspects of autonomous methodologies that generate an optimal grasp pose and orientation for a given object. Therefore, this thesis centres on an investigation of smart methodologies to identify novel objects, an effective method to accurately grasp an object, an investigation into the potential avenues of improving grasp rates and manipulation quality, and the development of a prototype that is driven by the proposed methods and methodologies. The scope of this study is limited to hardware, software and the conceptual process related to grasp synthesis. The outcome of the proposed system is detailed, and performance is quantified and qualified using various measures. This study is centred on 2-fingered, force-closure grasps for typical robotic arms. 3-fingered, 4-fingered, anthropomorphic and other end-effectors are beyond the scope of this research. RGB, RGBD, proximity and other sensors that are commonly used throughout industry are within the scope of this work, but more complex sensing apparatus, e.g., lidar, electrochemical and spectral sensors, are not. Electronic design, mechanical design and system integration are included as necessary.

The broader body of research—which this research is related to—deals with an ensemble of technologies aimed to increase the flexibility and customisation potential of industrial production systems that utilise robotics, AI, ML, sensing and automation. The goal of this research is to investigate a learning-based smart methodology for autonomously handling novel objects through the incorporation of part-related information and machine vision, considered from an industrial perspective. To fulfil this research aim, the following objectives must be accomplished:

1. To investigate current machine vision, digital image processing and object detection to identify potential techniques for object handling.
2. To study artificial intelligence, neural networks and machine learning pertaining to smart robotic manipulation.
3. To evaluate the potential of machine learning for novel part handling.
4. To research and identify the factors leading to better quality grasps.
5. To develop new methods/metrics to quantify the quality of a grasp.
6. To investigate the effect that grasp quality has on grasp rates.
7. To develop an AI-based part detection and grasp methodology that will enable robust, 2-fingered and autonomous novel object handling, while promoting future scalability and ease of integration.
8. To design a prototype system to physically validate the proposed object handling methodology and evaluate performance in terms of manipulation quality and grasp success rates.

1.6 Novel contribution

The novel contributions of this research are as follows:

1. *Grasp-induced error analysis*—an in-depth investigation of the factors that lead to reduced robotic manipulation quality. Subsequently, new grasp-induced error metrics are proposed to objectively quantify the error introduced by a robotic system when grasping an object. This error is measured by comparing an object's pre-grasp pose to its post-grasp pose.
2. *Post-grasp error minimisation*—the training methodology used to improve performance. By training regression models to predict the resultant grasp-induced error of a given sample, a selection process may be established to maximise for the proposed metrics; thus, improving manipulation quality and grasp rate.
3. *3-stage grasp synthesis framework*—the novel multi-stage process by which grasp locations and orientations are generated and selected. A novel generate-and-test structure is employed to sample the workspace for meaningful grasp poses. The resultant grasp pool is subjected to various selection criteria to finalise a pose for robotic implementation.
4. *Grasp synthesis that incorporates object properties*—the proposed method utilises object-property sensing hardware to improve grasp performance, e.g. force sensors for centre of mass acquisition. Many properties may serve as inputs to the selection stage if they are available, e.g., temperature, material properties, friction

coefficients, gripper feedback, etc. With the proliferation of IoT sensors, it has become relatively simple to measure many such properties.

5. *Small and efficient network structures*—relative to other CNNs throughout literature, the proposed networks are extremely small and computationally inexpensive. This is achieved by offloading complexity prior to network implementation, thereby simplifying the task. For example, objects are rotated prior to grasp detection to simplify the associated CNN, avoiding the need to train rotation-invariant models.
6. *Prototype system*—the prototype system utilised in this work was specifically developed to evaluate learning-based grasp synthesis methodologies. Consequently, the system may be used to assess a range of related approaches. Many novel calibration techniques were considered to improve the accuracy of the system and effectively integrate various subsystems, e.g., conveyor, bi-camera imaging system, robotic manipulator, load-cell coordinate system, etc.

The work in 1. and 2. resulted in a research paper outlining the utility of the proposed grasp-induced error metrics for improving the quality of grasp synthesis methodologies. A paper titled "Improving the quality of autonomous robotic grasps through the minimization of grasp-induced error" has been submitted to *IEEE Transactions on Robotics*, a journal with an impact factor of 6.123. This paper is currently under peer review. The 3-stage grasp synthesis framework in 3. was presented at the 2019 ICIEA conference in Xi'an, China. The outcome of the research work in 2., 3. and 4., is published in *IEEE Access*, an IEEE journal with an impact factor of 4.098. A paper detailing the design and related considerations in 6. was submitted under the title "A benchmarking platform for learning-based grasp synthesis methodologies" to the Elsevier *Robotics and Autonomous Systems* journal, which has an impact factor of 2.825.

Publications:

Paper title: A 3-stage Machine Learning-Based Novel Object Grasping Methodology
Journal: *IEEE Access*

J. J. V. Vuuren, L. Tang, I. Al-Bahadly, and K. M. Arif, "A 3-Stage Machine Learning-Based Novel Object Grasping Methodology," *IEEE Access*, vol. 8, pp. 74216-74236, 2020, doi: 10.1109/ACCESS.2020.2987341.

Paper title: Improving the quality of autonomous robotic grasps through the minimization of grasp-induced error
Journal (TBC): *IEEE Transactions on Robotics (T-RO)*

Paper title: A benchmarking platform for learning-based grasp synthesis methodologies
Journal (TBC): *Robotics and Autonomous systems*

Paper title: Towards the autonomous robotic gripping and handling of novel objects
Conference: *IEEE Conference on Industrial Electronics and Applications 2019*

J. J. v. Vuuren, L. Tang, I. Al-Bahadly, and K. Arif, "Towards the autonomous robotic gripping and handling of novel objects," presented at the *2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, June 19-21 2019, doi: 10.1109/ICIEA.2019.8833691.

1.7 Thesis Overview

This dissertation contains 11 chapters and three appendices.

Chapter 1 introduces the thesis by providing context. It discusses the motivation for the research, followed by an outline of its scope. The research problem is formulated in this chapter and a summary of novel contributions is presented.

Chapter 2 introduces background literature pertaining to the field of robotic object manipulation, citing topics such as artificial intelligence, machine vision, universal gripper design and grasp synthesis methodologies. Several issues surrounding a lack of community standardisation are noted, followed by common software tools used to facilitate research in this area. A conclusion is provided at the end of chapter 2.

Chapter 3 introduces grasp-induced error and offers an analysis of several common sources that lead to this error. This chapter also introduces the proposed similarity metrics used to quantify grasp quality.

Chapter 4 presents an overview and in-depth description of the proposed 3-stage methodology. A brief, qualitative description is provided, in addition to the mathematical basis of grasp pose generation and selection.

Chapter 5 establishes the pool of test objects used to assess the 3-stage methodology.

Chapter 6 pertains to the CNN used in the first stage of the proposed methodology, object classification. The derivation of the associated network architecture, training dataset and training properties is detailed, and the network selected for implementation is assessed.

Chapter 7 pertains to the CNN utilised by the second stage of the proposed methodology, grasp detection. Details related to the CNN used in the second stage are provided in this chapter. Several network architectures, dataset configurations and other properties are considered prior to training and network deployment.

Chapter 8 pertains to the third stage of the proposed methodology, grasp selection. This chapter details the samples and methods used to train several regression networks to predict the outcome of a candidate grasp pose.

Chapter 9 details several simulations conducted prior to constructing the prototype system. Hardware, software, vision, coordinate-frame consolidation and calibration considerations surrounding the system are also noted in this chapter. The experimental protocol used to trial the method is discussed in chapter 9.

Chapter 10 presents a qualitative and quantitative analysis of the results of trialling the proposed methodology 4,000 times over four rounds of testing. The results and their implications are further discussed in this chapter.

Chapter 11 contains the main conclusions of this research and recommendations for future work. A brief case study is noted and the location of an online depository containing files related to this research is provided.

Appendix A describes a brief industry perspective survey related to an initial proposal of this work. Several industries were contacted to gauge general interest in a novel object handling methodology within their related fields.

Appendix B contains a photograph taken at the 2019 ICIEA conference in Xi'an, China, where this work was presented.

Appendix C tabulates the results of the second round of trials.

Appendix D contains a brief glossary of terms used throughout this thesis.

Chapter 2

Literature review

2.1 Artificial intelligence

Within computer science, AI refers to the study of techniques, algorithms and statistical models that rely on patterns and inference to perform specific tasks, as opposed to explicit instruction. This term was coined by John McCarthy in a conjoined study proposal in 1955 [133]. Many implementations of this technology have been iconic in defining AI for the public. IBM's Deep Blue for instance, left a lasting impression in 1997 after winning a 6-game chess rematch against—then world champion—Gary Kasparov [134, 135]. In 2011, IBM's Watson competed on *Jeopardy!* and won against two of the game's best players [136]. Their system has since been utilised as a knowledge base supplement in healthcare [137, 138]. In 2016, Google's AlphaGo Lee won four out of five games against Korean Go Champion Lee Se-dol [139] and in 2017, Google's AlphaGo Master defeated 60 professional Go players online and won three consecutive matches against Chinese Go world champion Ke Jie [140]. Their system used self-play reinforcement learning and required no human interaction during its 40-day training period. AI is more commonly used in commercial applications, e.g., facial expression recognition [141], quality assessment of live video streaming [142], smart assistants such as Alexa and Hello Google [143], moderating unwanted players in video games [144], email spam filtering [145], self-driving car technology [146, 147], online customer support [148], consumer behaviour prediction [149], the detection of botnets online [150] and the prediction of foreign exchange rates [151]. AI is also prominent in vision-related research, often studied in segmentation [22, 152-157] and object recognition [105, 158-161].

The application of AI has surged in popularity in many fields over the past two decades, facilitated by major advances in capability and the lowering cost of graphics processing units (GPU) [134, 135, 162, 163]. The 1999 NVIDIA Riva TNT2 for instance, was capable of 0.25 billion (giga) floating point operations per second (GFLOP). The 2019 NVIDIA RTX 2060 Super represents a price equivalent GPU, capable of 7,180 GFLOPs. GPU processing capability and cost trends are illustrated in Figure 2.1 and Figure 2.2. This trend has made several aspects of ML—such as deep learning using artificial neural networks and large amounts of data—more practical and affordable.

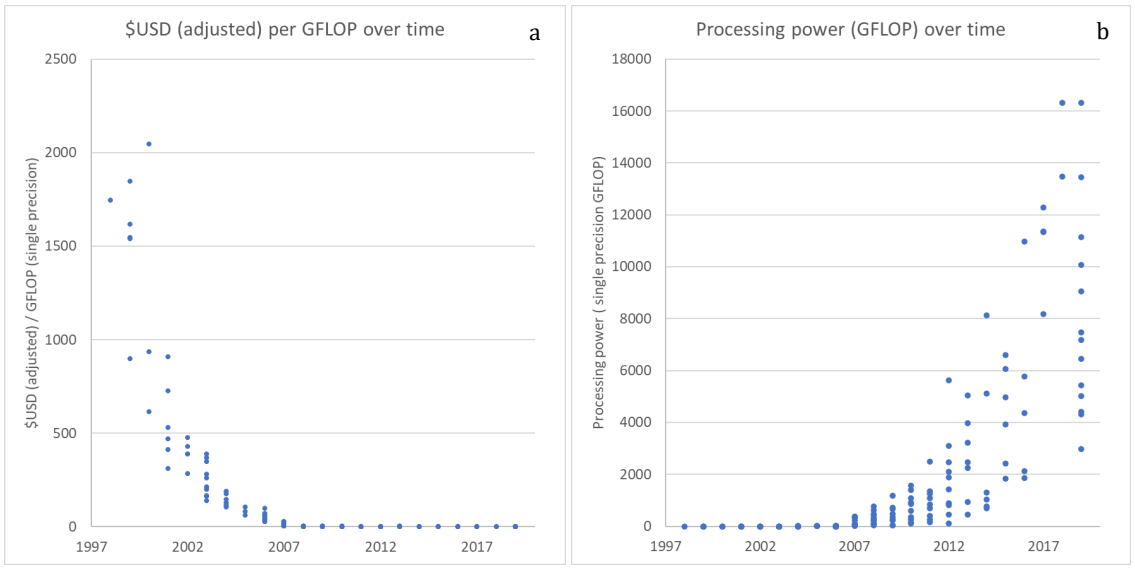


***Figure 2.1*. (a)—adjusted release price (USD) per GFLOP (single precision) of processing power of consumer desktop NVIDIA GPUs 1998 - 2019. (b)—processing power in GFLOPs (single precision) of desktop NVIDIA GPUs available to consumers 1998 - 2019. The information required to collate these graphs was obtained from various sources [164-175].**

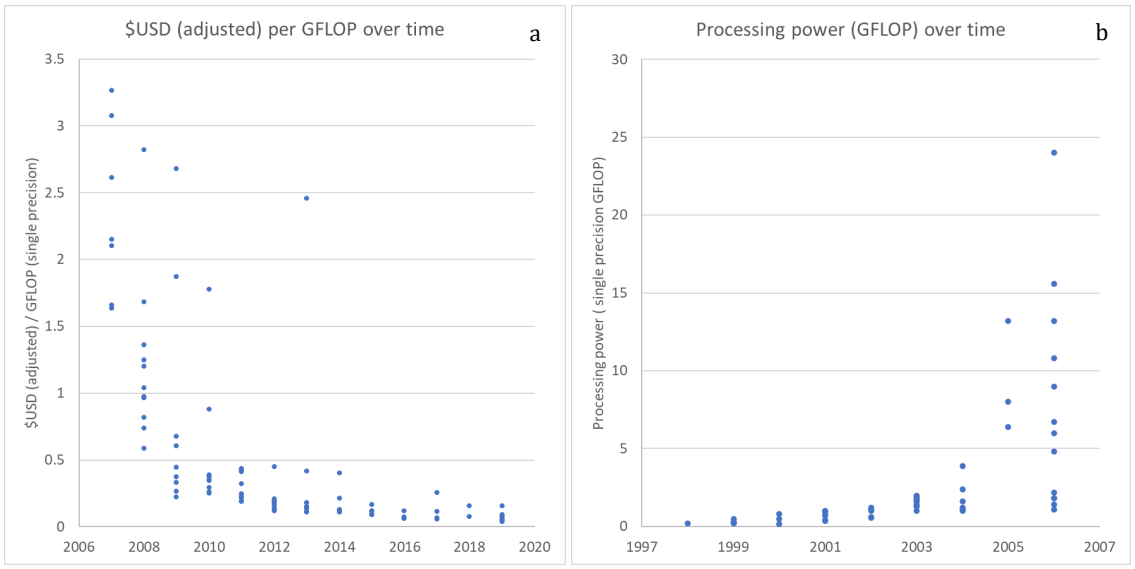


***Figure 2.2*. (a)—adjusted release price (USD) per GFLOP (single precision) of processing power of consumer desktop NVIDIA GPUs 2007 - 2019. (b)—processing power in GFLOPs (single precision) of desktop NVIDIA GPUs available to consumers 1998 - 2006. The information required to collate these graphs was obtained from various sources [164-175].**

Compared to the memory access speeds between CPU and RAM, GPUs offer high memory band-width that allows for thread scheduling with very little overhead [163]. ML tasks benefit from such parallel computing structures. NVIDIA hardware is overwhelmingly prevalent in this context because of their proprietary compute unified device architecture (CUDA) parallel computing platform. Many popular libraries within the ML community exploit the CUDA architecture, e.g., TensorFlow [176], Caffe [177], PyTorch [178], Keras [179], CNTK [180], Theano [181] and the Matlab Parallel Computing Toolbox [182, 183].

The growing use of AI remains a global trend and is reflected by the amount of investment in related business ventures. Analysis by Quid suggests that in 2015, US $8.5 billion was spent on AI start-ups—nearly four times as much as 2010 [184]. According to CB Insights,

over 3,600 AI start-up companies in over 70 countries have raised US $66 billion since 2013 [185]. Figure 2.3 shows the estimated funding of AI start-ups worldwide since 2013. Moreover, Allied Market Research expects the global AI market size to exceed US $169 billion by 2025 [186]. ML is forecasted to show the largest growth, followed by natural language processing, image processing and speech recognition. Fortune Business Insights valued the global market size of AI at US $20.7 billion in 2018 and project that this value will increase to over US $202 billion by 2026 [187]. This trend is also shared by New Zealand (NZ). The NZ-based company FaceMe for instance, has recently completed a NZ $15 million funding round [188]. Currently, more than 200 companies are working on AI in NZ [189] and by 2035, AI is predicted to increase the NZ GDP by up to NZ $54 billion [190].

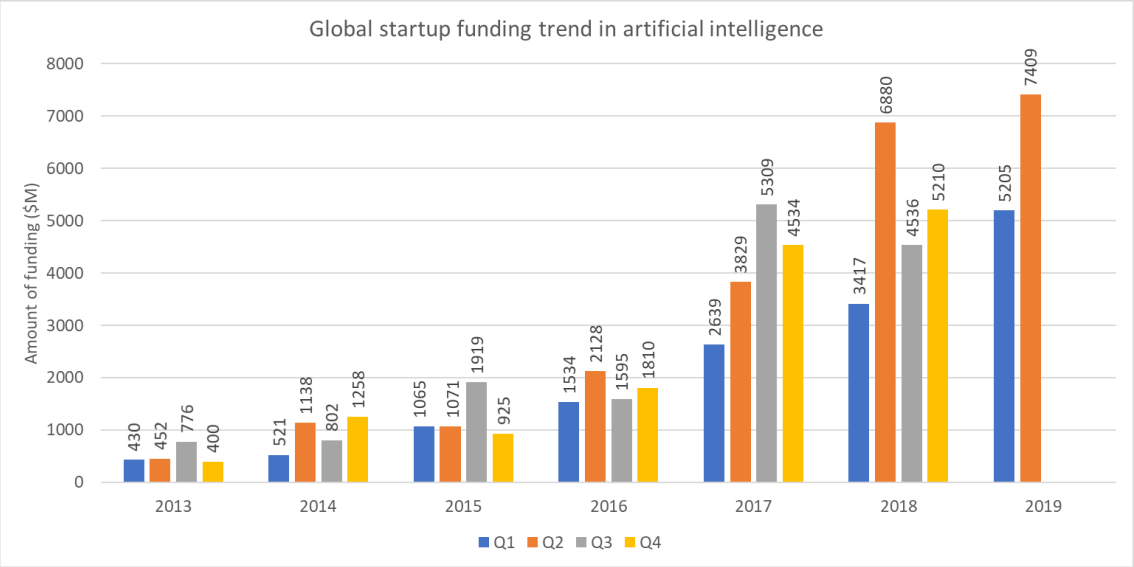


***Figure 2.3*. Estimated global funding of AI start-ups. Source: CB Insights [185].**

AI is a parent term that refers to many subfields. This is illustrated in Figure 2.4—a. Deep learning for example, is a type of ML that represents an input as a nested hierarchy of abstractions, each defined in relation to less abstract representations. Figure 2.4—b illustrates this graphically. Some proponents in the field posit that ML is the only viable approach to achieving AI systems that can adequately deal with the inherent uncertainty related to real-world problems [135, 191]. This is supported by the recent success of ML in many fields. Prior to the application of deep learning in 2012 for instance, ImageNet large scale visual recognition challenge (ILSVRC) results were improving slowly, with top-5 error rates of 28.2% [192] and 25.7% [193] in 2010 and 2011, respectively. In 2012, the winning CNN-based model AlexNet achieved a top-5 error rate of 15.3% [104], compared to 26.2% achieved by the second-best entry [194]. Since the introduction of deep learning in ILSVRC, competition error rates have fallen dramatically to 14.7% in 2013 [195], 6.7% in 2014 [196], 3.6% in 2015 [197], 3.1% in 2016 [198] and 2.25% in 2017 [199]. The human error rate sits at approximately 5.0%. Other forms of AI not under the branch of ML have also been well-studied, e.g., semantic networks [200, 201], rule-based expert systems [202], description matching and goal trees[203], rule-chaining, problem reduction and search methodologies [204, 205], although these topics have not attracted much attention recently.

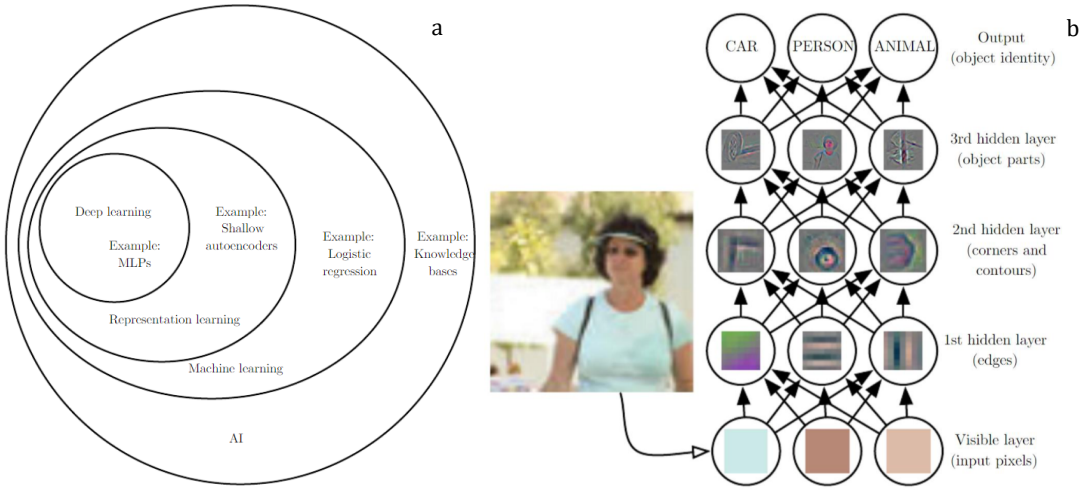


***Figure 2.4*. (a)—Venn diagram of various AI approaches and subfields. (b)—illustration of the hidden representations used by a deep learning model to interpret an image. Source: Goodfellow, Bengio and Courville [135].**

ML techniques are broadly classified into three categories: supervised learning, unsupervised learning and reinforcement learning [135, 206, 207]. In supervised learning schemes, the goal is to learn a model from a fixed dataset, wherein each sample has associated features and a label or target. This model can then be used to make predictions about unseen or future data. Unsupervised learning algorithms use unlabelled datasets containing many features to learn useful properties about the structure of the data. Reinforcement learning algorithms do not operate on a fixed dataset. Such algorithms improve their performance based on interactions with an environment, generally in relation to some reward system. Further subcategories of ML are presented in Figure 2.5. Classification is a very popular branch of supervised learning, in which an algorithm assigns a known category to a given sample. Handwritten digit classification is a classic example [208]. In contrast to the distinct categorisation of samples, regression algorithms are used to predict a continuous output. Wao, Bivins, Hunt, Ries and Schattner for instance, use SAT and ACT scores to predict the undergraduate GPA scores of construction science and management students [209]. Clustering is a popular subcategory of unsupervised learning, and has been used to infer population structures within human genetic data [210].

Artificial neural networks (ANNs) are connected systems inspired by biological neural networks that learn to perform tasks by considering samples. ANNs fall under deep learning and can be trained via supervised, unsupervised or reinforcement schemes. A comprehensive explanation of ML theory is not within the scope of this thesis. For a practical guide to ML, the reader is referred to the Python machine learning resource by Raschka and Mirjalili [206]. For a more mathematical basis of ML, the reader is directed to the resource by Goodfellow, Bengio and Courville [135].

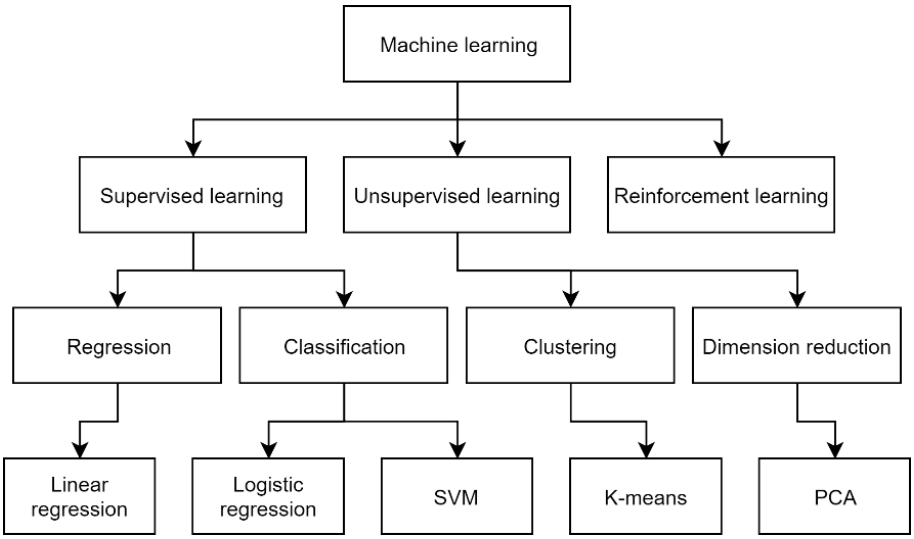


***Figure 2.5*. Categorisation hierarchy of machine learning models. Source: Dasila [211].**

ML has significantly impacted research related to industrial control systems, automation and robotics. Yao, Chow, You and Chan for instance, use support vector machines (SVMs) to detect anomalous behaviours of programmable logic controllers (PLCs) [212]. Kanawaday and Sane use ML to predict the maintenance requirements of industrial machines using internet of things (IoT) sensor data [213]. Amihai *et al.* show that vibration data and ML can be used to predict the condition of industrial assets [214]. Various ML techniques are also often used in robotic manipulation research, e.g., SVMs [33, 62], binary classifiers [107], K-nearest neighbours (KNNs) [83], Gaussian process classification [82], ANNs [47, 53, 54], adversarial learning [61], deep reinforcement learning [11, 36, 76] and CNNs [18, 19, 40-42, 71, 75, 77, 85]. ML is used for many aspects of grasping. Common tasks include object and pose detection [17-19, 21] and grasp synthesis [58, 69, 71, 73]. Chapter 2.5 provides more information regarding manipulation literature.

This thesis investigates the employment of AI for various tasks related to automated grasping. ML has several advantages over traditional approaches. First, ML brings more robustness to noise and provides better generalised performance. In a paper published by Miyazaki and Miura in 2017 for example, 71% grasp rates for pre-defined, distinctly coloured rectangular blocks in clutter were achieved algorithmically using RGBD sensor data [8]. In the same year, Viereck, ten Pas, Saenko and Platt achieved 89% grasp rates for general household objects in clutter using CNNs and RGBD data [75]. The increase in generalised performance associated with ML has also been seen in object recognition literature, quantified by the ILSVRC competition results [198]. Second, ML models can generally be executed in less time than traditional processes. Continuing with the previous comparison, the algorithmic approach [8] took 43-66 seconds to determine the target object and location. In contrast, the CNN-based system [75] operated at approximately 5 Hz. Similar systems have been shown to operate at very high frame rates when GPU technology is employed. Fast YOLO for instance, operates at 155 Hz on a Titan X GPU [160]. This system simultaneously locates and classifies multiple objects within a scene.

ML also has significant disadvantages. Within manipulation literature, it is common to exceed a training dataset size of 20,000 samples [14, 18, 21, 33, 60, 61], which may require significant resources and time to collect. Depending on the desired degree of generalisation and type of learning employed, some training sets can exceed 500,000 samples [35, 36, 75].

Levine, Pastor, Krizhevsky and Quillen for example, collected 800,000 training samples over the course of two months using between six and 14 robotic manipulators [35]. Moreover, a lack of intra-dataset variation harshly reduces the generalisation performance of an ML model. Conversely, it may be very difficult to form any useful generalisations about a dataset if there is too much variation. The time and resources needed to train ML models can also be problematic. Caldera, Rassau and Chai for instance, required over 32 hours to train a single model for grasp detection using a Titan X GPU [73]. Training time can become a real constraint, as numerous models may need to be trained before converging on a suitable network architecture for the application. Morrison, Corke and Leitner trained 95 grasp synthesis CNNs of different structures prior to their final implementation [72]. Several dedicated cloud services that specialise in AI training have emerged to address this issue. Google Cloud for example, provides GPU-enabled training services online [215]. The lack of interpretability of AI is another well-known issue within the ML community [216-219]. This is particularly relevant in applications such as self-driving cars [147], where it is important to understand why a model makes a certain prediction and how to manipulate this prediction in the future.

2.2 Machine vision

Appearing as a subcategory of computer vision in the 1970s [220-222], the term machine vision (MV) is generally used to describe a digital image processing (DIP)-based system that has been adapted to provide usable information for hardware application. It should be noted that there is a slight distinction between DIP—in which both input and output are images—and MV. MV aims to extract very specific information from an image, often in conjunction with other sensing technologies. MV is frequently used in industry and academia. Some studied applications include fruit sorting [223] (Figure 2.6), general inspection and quality control [224-226], automated sorting [227-229], potato chip inspection [230], coal sorting [231], milk powder analysis and grading [232] and e-waste sorting for electronic recycling [233].

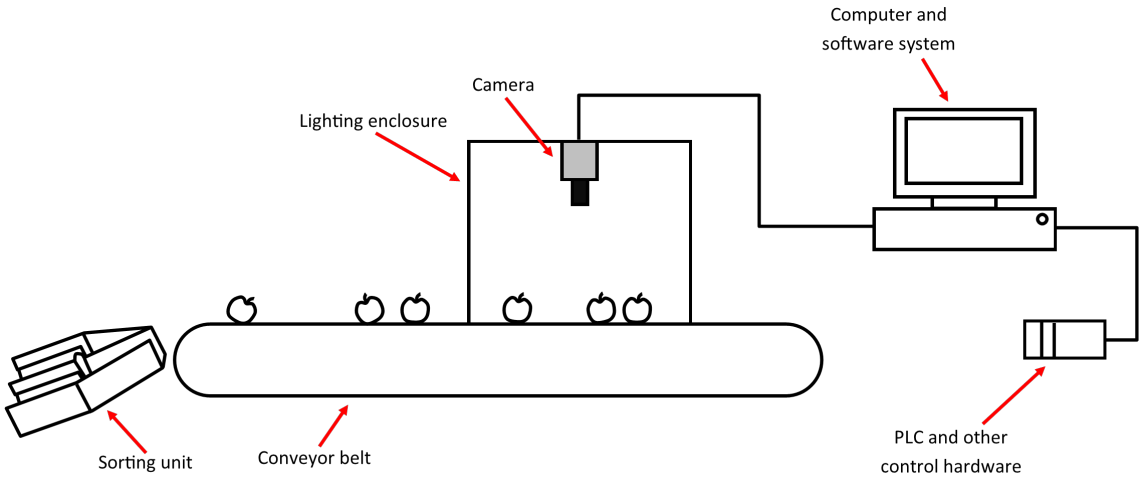


Figure 2.6. **Schematic representation of a machine vision system for apple sorting.**

Processes that employ MV often implement some form of external light control to provide a more consistent inspection environment—particularly for industrial application. Light consistency reduces noise, thus increasing the reliability and quality of a vision system

[224]. Milioto, Lottes and Stachniss for instance, use an enclosure to isolate their crop inspection area from the varying degrees of outdoor light [152]. Artificial light is used to maintain consistent lighting. Stein, Bargoti and Underwood combine a colour camera and strobe to increase the reliability of fruit detection in an orchard block for yield estimation [234]. Laszlo, Holonec, Copîndean and Dragan control the lighting and background of their proposed conveyor inspection system to more accurately determine the size and shape of e-waste [233]. Many other works employ similar techniques to increase the robustness of their MV-based systems [223, 228, 230, 235]. A full description of the plethora of techniques used in MV lighting is beyond the scope of this thesis. For a practical guide, the reader is referred to the resource by Advanced Illumination [224]. This source suggests that—if applicable—light control via an enclosure provides the most consistent inspection environment. External light control and background uniformity is also common in manipulation research [66, 73, 78, 85]. Mahler *et al.* for example, use a green background and artificial light to maximise object contrast for their proposed grasp synthesis methodology [40]. Similarly, Harada *et al.* maximise object contrast with a blue background [56].

As a matter of course, it is often useful or necessary to separate an object or region of interest within an image from the rest of the scene for further processing or interpretation. The term segmentation describes a variety of related techniques used to divide or partition an image. Thresholding is a very popular form of segmentation [236, 237]. In global binary thresholding for example, each pixel within an image is replaced with a black pixel if the image intensity value $I[x, y]$ is below some fixed constant $Thresh$, or a white pixel if the image intensity value is equal or greater than $Thresh$—enumerated in Equation 2.1. This creates a new binary image I_{Thresh}:

$$I_{Thresh}[x, y] = \begin{cases} 1, & if\ I[x, y] \geq Thresh \\ 0, & if\ I[x, y] < Thresh \end{cases} \quad \textbf{(2.1)}$$

where x and y range from one to the width and height of an image, respectively. A graphical representation of the outcome of this procedure is shown in Figure 2.7.

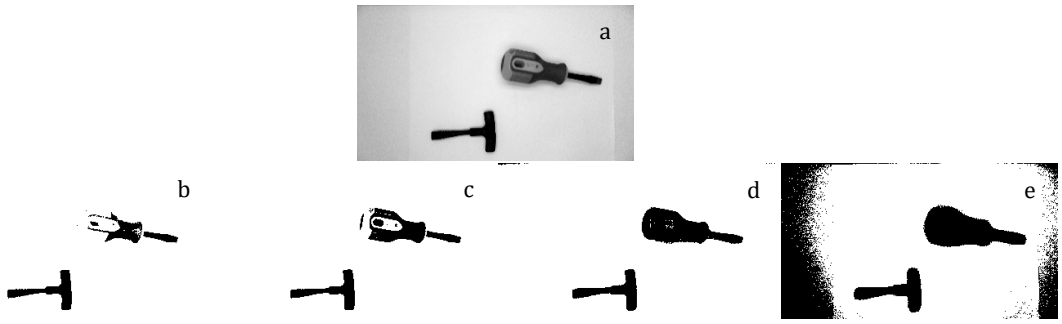


***Figure 2.7*. (a)—original greyscale image. (b)—global threshold of 50 applied. (c)—global threshold of 100 applied. (d)—global threshold of 150 applied. (e)—global threshold of 200 applied.**

This basic method has been used to effectively segment milk powder particles [232] (Figure 2.8—a, b), segment potato chips [230] (Figure 2.8—c), locate objects based on colour [229] and detect and classify moving objects for industrial quality control [238]. Otsu's method is a popular thresholding algorithm used to automatically find the global threshold constant $Thresh$ that maximises the contrast between object and background by maximising the variance between local histogram maxima [239]. Sezgin and Sankur provide a useful

overview and comparison of traditional thresholding techniques [240]. Image segmentation is an extremely broad subfield under computer vision. Once segmented, a plethora of morphological filters are available to further refine the image. For a brief introduction to common segmentation techniques, the reader is referred to the overview by Yuheng and Hao [237].

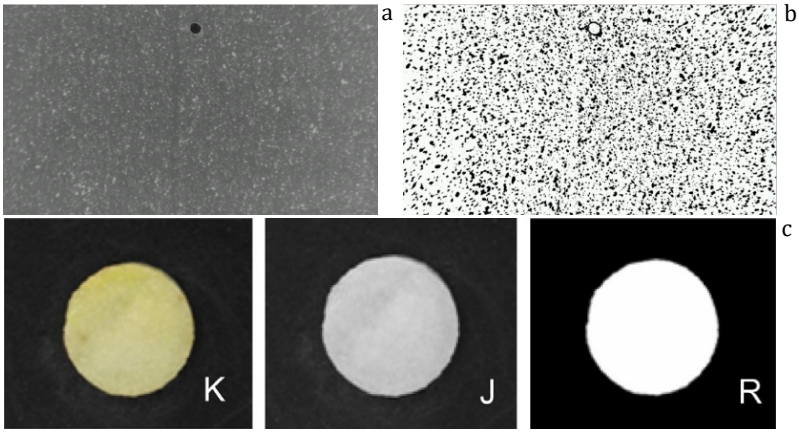


Figure 2.8. **(a)—greyscale image of milk powder [232]. (b)—binary image of milk powder, obtained via global threshold [232]. (c)—potato chip segmentation [230].**

Image segmentation is often a precursor to object detection and/or recognition. Traditional object recognition techniques centre around image features. Features are useful for object detection and recognition because they allow images to be described and compared in terms of features—as opposed to pixel intensity. A very popular method of multi-scale, perspective/rotation-invariant feature description is the scale-invariant feature transform (SIFT), first proposed by David Lowe in 1999 [241]. Other common feature descriptors and detectors include the Harris & Stephens edge and corner detector [242], the features from accelerated segment test (FAST) feature point extractor [243], the speeded up robust features (SURF) detector and descriptor [244], Canny edge detection [245] and the Sobel-Feldman operator for edge detection [246]. A comprehensive description of DIP methods is beyond the scope of this thesis. Woods and Gonzales provide a thorough resource that describes the mathematical basis of DIP, and their latest edition (2018) touches on the fundamentals of ML for vision [247]. For a more basic overview, the reader is referred to the introduction to machine vision by Cognex [225], or for a more practical perspective on MV, the resource by Szeliski [220].

ML has become somewhat of the de facto methodology for image classification and object recognition tasks for both industry and research. Lightweight IoT devices built around AI computing for instance, have gained significant traction within industry and the consumer market. The NVIDIA Jetson Nano for example, contains 128 CUDA cores capable of 472 GFLOPs, while only consuming 5 W [248]. Google's Coral Edge is another example currently in development [249]. ML has demonstrated good performance for classification tasks within industry. The EAVISE Research Group for instance, trained a linear binary SVM classifier for Orchid species categorisation within an industrial setting [250]. Their classifier showed good performance using five manually defined features and a small dataset containing only 360 samples. Although ML classifiers can operate using features that have been manually defined, the major advantage and convenience of such algorithms stem from their ability to generate their own features. Moreover, ML techniques refine such features during training. Generally, this distinction constitutes the difference between deep learning

methodologies and traditional ML. Other studied vision classification tasks include the classification of land cover (wheat, maize, soybeans, etc.) [157], facial expression classification for mood recognition [141], military vehicle recognition from SAR images [161], classification of sonar images [251], classification of Arabic numerals [252] and real-time quality assessment of live video streaming [142]. As mentioned in Section 2.1, ML has revolutionised the field of image object recognition—as partly evidenced by the improvements brought to the ILSVRC competition results by ML [198]. In a recent series of papers [158-160], the you only look once (YOLO) architecture has demonstrated exceptional real-time performance in simultaneous localisation and classification of objects. By modelling detection as a regression problem, their framework predicts a bounding box for an object and the object within that box. The base model processed images at 45 FPS, while a smaller version operated at 155 FPS [160].

Image segmentation methodologies based on ML have also been studied, e.g., pixel-wise semantic segmentation and classification [153], multi-class object segmentation for the APC [22], biomedical segmentation for cell-tracking [154] and general pixel-wise semantic segmentation for scene understanding [105, 155, 156]. Though ML can be used to enhance segmentation, traditional DIP techniques are often used for pre-processing. Milioto, Lottes and Stachniss for example, transform a conventional RGB image into 14 unique representations, e.g., red component, green component, Sobel in x-direction, Canny edge detection, HSV colour space, etc. (Figure 2.9—b, c, d, e) [152]. All representations are then fed into a CNN which classifies each pixel as either value crop, dirt or weed. By masking this classification over the original image, their agricultural platform can accurately detect the presence and location of weeds and value crop [152]. The outcome of this process is illustrated in Figure 2.9—f. It should be noted that their work observed a 19% increase in pixel-wise classification rates when implementing their alternate representations, compared to RGB data only.

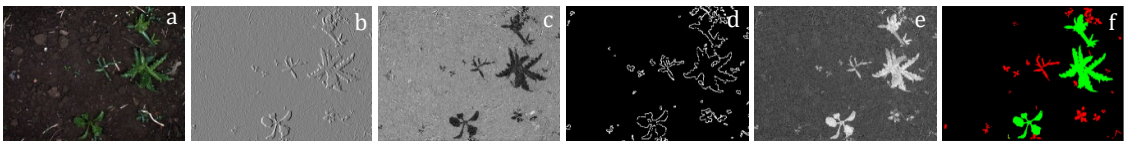


***Figure 2.9*. Illustration of the various representations used by a CNN to classify value crop. (a)—original RGB image captured by system. (b), (c), (d), (e)—various alternate representations of the RGB image. (f)—classification mask where red represents the weed class and green represents the value crop class. Source: Milioto, Lottes and Stachniss [152].**

This research studies the potential of machine vision for object identification, object pose estimation and grasp detection. MV, DIP, image segmentation and object recognition technologies often form the basis of automated robotic manipulation. Conventional RGB cameras are commonly employed [14, 23, 35, 36, 59-61, 77], although depth cameras (RGBD) have been a significant research focus in recent years [11, 12, 17, 20, 66, 70, 73, 76]. Other input modalities such as laser sensors [27], infrared scanners [31] and lidar and optical proximity sensors [253] have also been suggested. The methodology proposed in this thesis studies two monocular RGB observations of the scene, captured within an enclosure. Situating an imaging system within an enclosure provides two major advantages. First, an enclosure provides a consistent inspection environment [224]. Second, a well-defined background provides good general object contrast—simplifying the segmentation process [237]. Moreover, enclosures are already commonly coupled with industrial robotic systems utilising vision. Although RGBD sensors have several advantages over RGB, they are not incorporated in this work for several reasons. RGBD sensors generally require a

distance in excess of one metre for accurate measurements [254, 255]—which make them impractical for enclosure implementation. Generally, the precision of depth sensors sits between 30-60 mm [255], which is too large for accurate manipulation within small-part assembly applications. Additionally, some reflective and transparent surfaces cannot be captured by modern depth sensors [256]. Industrial processes generally require systems that are real-time. In ML theory, it is well-known that a reduction in the input dimensionality of ML algorithms significantly improves computational efficiency [206]. RGBD data generally consists of 7 channels, compared with the 3 channels associated with RGB data [13]. For an explanation of how RGBD sensors acquire data, the reader is referred to the work by Zollhöfer [257].

2.3 Universal gripper designs

Early novel object grasping approaches centred around physical design. A journal paper published in 1985 for instance, proposed a pin-array structure consisting of numerous sliding pins that conform to object shape and size [258]. This design is illustrated in Figure 2.10—a. Novel 3-fingered designs were also common throughout this period. Van der Loos proposed a new 3-fingered design for industry in 1978 [259]. NASA expanded on this design in 1989 by adding tactile feedback [260]. Many later works focused on anthropomorphic designs for dexterous grasping. The DIST-hand for example, was first proposed in 1998 [261]. This design was based on the human hand and actuation was achieved via a tendon design (Figure 2.10—b) [261, 262]. For a useful overview of legacy anthropomorphic end-effectors, refer to Hollerbach and Jacobsen [263]. 2-fingered and 3-fingered grippers were also popular in this era. Causey and Quinn for example, proposed guidelines for 2-fingered gripper design within modular manufacturing (Figure 2.10—c) [264]. Laliberté, Birgle and Gosselin proposed an underactuated 3-fingered gripper with moveable joints (Figure 2.10—d) [48]. Their design was shown to conform to object shape and demonstrated a lifting capacity of 70 kg. Approaching novel object grasping from a design perspective is still prevalent today.

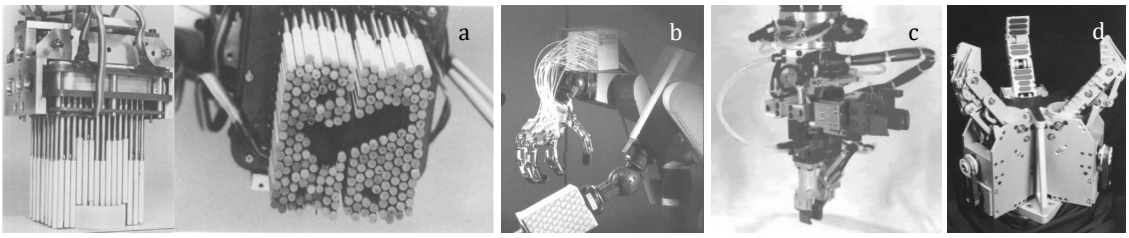


***Figure 2.10*. (a)—sliding pin-array design proposed in 1985, dubbed the Omnigripper [258]. (b)—the anthropomorphic DIST-hand first proposed in 1998 [262]. (c)—2-fingered modular design proposed in 1998 [264]. (d)—underactuated 3-fingered design with moveable joints proposed in 2002 [48].**

Many new designs and revisions are still being proposed in contemporary works, particularly in the form of 2-fingered, 3-fingered and 4-fingered force-closure designs. Chan and Cheung for instance, proposed a small 2-fingered design that makes use of variable reluctance actuators that do not employ permanent magnets [265]. Their design is illustrated in Figure 2.11—a. Wang, DelPreto, Bhattacharyya, Weisz and Allen proposed the Columbia Hand (Figure 2.11—b, c)—a 9-DOF end-effector consisting of two fingers, one thumb, 10 angle sensors and nine force sensors [45]. They achieved stable grasps for various objects by conforming to the object itself based on sensory data, which was also used to predict and adjust for grasp quality. An example of their design conforming to object

shape is given in Figure 2.11—c. Spiliotopoulos, Michalos and Makris propose a similar 3-fingered design for flexible assembly automation [266], while Shauri, Salleh and Hadi apply their design to general household grasping [49]. Yao, Zhan, Ceccarelli, Carbone and Lu investigate the underlying grasp strategy and control methodology of 3-fingered robotic hands [47]. The 4-fingered SDM hand is paradigmatic in the context of grasping novel objects through design. Proposed by Dollar and Howe [50], the SDM hand (Figure 2.11—d, e) aims to accommodate for the uncertainty in grasping tasks by incorporating features such as compliance and adaptability. Their manipulator employed viscoelastic flexure joints and a nylon-coated cable structure to grasp objects in the presence of significant positional error. Their experiments showed good grasp performance for positioning errors of up to 100% of the object size at the 5 cm scale and 33% at the 10 cm scale [51]. Later, they implemented a minimalistic control methodology based on colour segmentation [52]. Despite the simplistic nature of their control, their gripper demonstrated good performance for a range of common household objects. An example of the SDM hand grasping a wine glass can be seen in Figure 2.11—e.

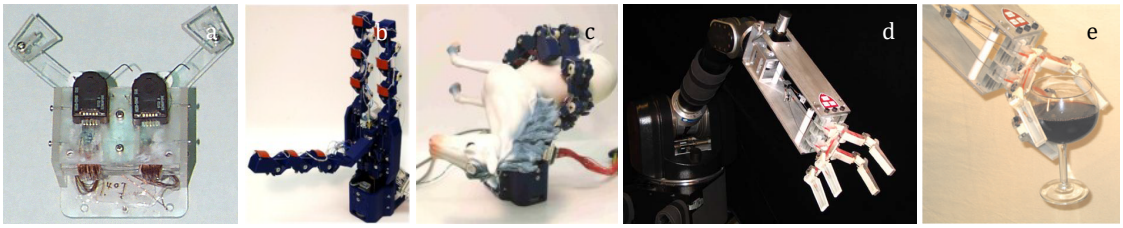


***Figure 2.11*. (a)—2-fingered, variable reluctance gripper design [265]. (b), (c)—the 3-fingered Columbia Hand in an open position, or grasping a toy [45]. (d), (e)—the 4-fingered SDM hand in an open position, or grasping a wine glass [50-52].**

Anthropomorphic end-effectors have remained a significant topic of interest for many works [267-273]. Gaiser *et al.* for example, proposed the 11-DOF FRH-4 humanoid hand [274]. Their design is actuated by flexible fluidic actuators to promote safe interaction with human operators. Similarly, Polishchuk presented a 5-fingered gripper actuated by pneumatic chambers mounted on spherical hinges [275]. This design focused on novel object handling for industrial implementation. Welhenge, Wijesinghe and Rajakarun concentrated on the finger structure design of a gripper that emulates human finger motion [34]. Mudigonda, Agrawal, Deweese and Malik investigated deep reinforcement learning for grasping novel objects with anthropomorphic hands [276]. They demonstrated good performance in simulation. Other works have also studied control within this context [53, 54]. Figure 2.12 illustrates various examples of popular anthropomorphic hands throughout literature.

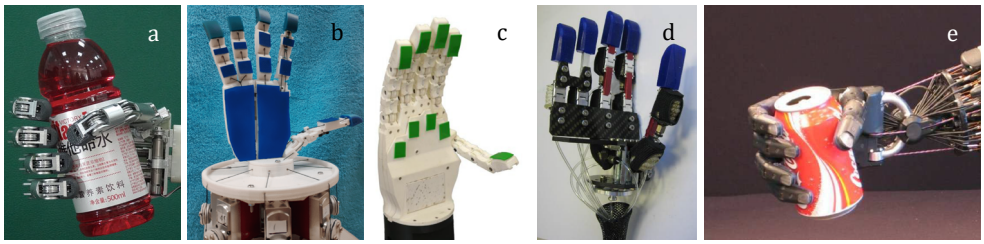


***Figure 2.12*. (a)—the X-hand grasping a filled bottle [267]. (b)—the 19-DOF CATCH-919 hand [270]. (c)—the 15-DOF ARMAR hand [271]. (d)—the 11-DOF FRH-4 hand [274]. (e)—the Shadow dexterous hand grasping an empty can [54].**

While general control has improved, the effective autonomous control of complex manipulators remains a significant barrier to pragmatic implementation. Some works for

example, require human collaboration. The work of Huang, Lehman, Mok, Miikkulainen and Sentis illustrates this. They proposed a semi-autonomous grasp methodology for a 5-fingered gripper, in which a bounding box is manually defined around the target object prior to grasp synthesis [277]. The location and configuration of a grasp for complicated end-effectors can be particularly difficult to represent conceptually, further complicating control theory, although this has been addressed by some works. Weiss and Vincze for example, present a universal grasping methodology, trialled on household objects for four varying grippers [278]. Grasp representation is discussed in detail in Section 2.4, and the associated synthesis methodologies in Section 2.5. In recent years, there have been many attempts to address the issue of control difficulty through design simplification by means of under-actuation [44, 46, 48, 79, 279-281]—where the number of actuators is less than the number of degrees of freedom. Generally, this design principle yields end-effectors that conform to the target object shape. A variety of alternate design-based avenues have also been explored, e.g., meshed pin-array structured end-effectors similar to the Omnigripper (Figure 2.13—a) [282-284], a circular variant of the meshed pin-array design [285], universal granular jamming end-effectors (Figure 2.13—b) [55, 286-288], a fluidic solidification-based end-effector that solidifies under a magnetic field [289], nubbed granular jamming end-effectors [290], a highly articulate tubular end-effector (Figure 2.13—c) [291], an end-effector that utilises both 3-fingered force-closure and granular jamming [292] and soft pneumatic grippers (Figure 2.13—d) [293, 294].

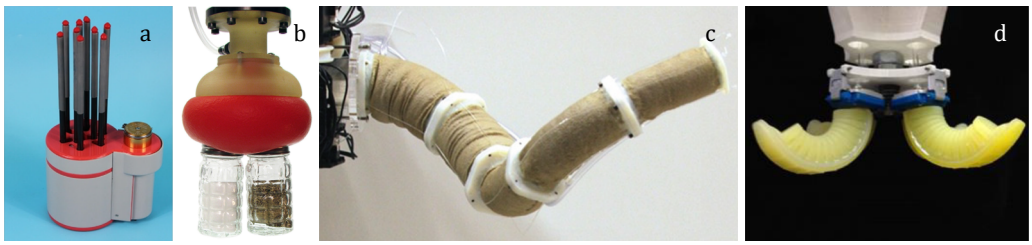


***Figure 2.13*. (a)—meshed pin-array structured end-effector [282, 284]. (b)—universal gripper based on granular jamming [286]. (c)—highly articulate tubular gripper design [291]. (d)—soft pneumatic robotic gripper design [293].**

This research studies 2-fingered, force-closure end-effectors. Grasping novel objects based on gripper design has shown promise, though there are several current issues associated with this approach. Generally, convoluted grippers excel at specific tasks to the detriment of generalised performance. This was particularly evident in the trials submitted by Jiang, Amend, Lipson and Saxena [288]. They compared the performance of a granular jamming gripper to a traditional 2-fingered gripper. Of the 23-object test pool, eight objects were not graspable by the granular end-effector, while the 2-fingered gripper could not grasp four objects [288]. Moreover, the 2-fingered gripper showed better grasp rates for objects that were graspable by both end-effectors. It should be noted that an identical control methodology was used to generate grasps for both types. The granular jamming gripper proposed by Brown *et al.* struggled to manipulate flat, or approximately flat objects [55]. They also showed very poor performance in terms of holding force for various shapes. Of the eight objects tested in their trial, five resulted in a holding force of less than 5 N, and one object was not graspable [55].

In many circumstances, the ability of a gripper to successfully lift an object is of equal importance to the subsequent peri-grasp manipulation quality and post-grasp placement quality. The under-actuated design proposed by Odhner, Ma and Dollar for example, excelled at grasping very thin objects from flat surfaces [44]—which is difficult for many contemporary works. Though their design showed good performance in terms of initial

contact, it generally grasped objects such that placement accuracy was compromised. An example of this is shown in Figure 2.14—a. The same issue was present for meshed pin-array structured end-effectors (Figure 2.14—b), granular jamming end-effectors (Figure 2.14—c) and the SDM hand (Figure 2.14—d). The SDM hand generally forced objects into some gripper-relative position, shifting the object relative to the original contact configuration [50-52]. This shift becomes problematic, as the precise orientation of the object within the gripper is not known, which can introduce error on placement. This observed error is a result of the interaction between the gripping interface and the object—and can be measured. In this thesis, the error introduced by this interaction is studied for one gripper type, although this measurement is applicable regardless of gripper configuration.

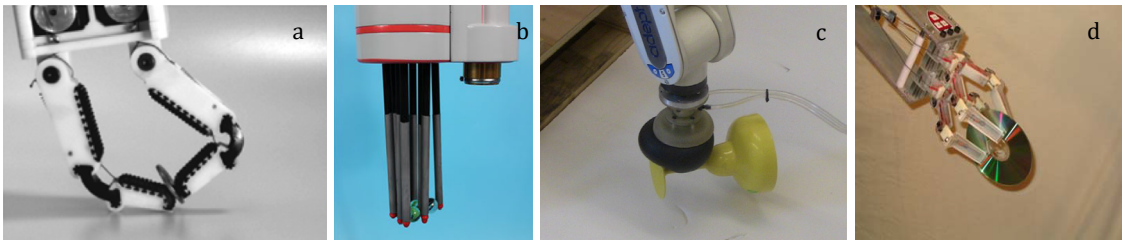


Figure 2.14. **(a)—an under-actuated 2-fingered design grasping a coin [44]. (b)—the CTSA hand grasping a small toy [284]. (c)—a universal jamming gripper grasping a plastic goblet [288]. (d)—the SDM hand grasping a CD [50-52]. These examples illustrate an issue with complex gripper designs. As shown, objects are grasped such that subsequently placing the object accurately is difficult.**

The above-mentioned issues are particularly relevant for industrial assembly, where many situations demand the controlled gripping, handling and placement of parts for subsequent processes. Consequently, the type of grippers utilised by industry and autonomous object manipulation research has largely centred around parallel jaw, force-closure grasps [56-62, 70-72, 74-77, 84]. Moreover, many works considered to be state-of-the-art implement 2-fingered grippers [12, 14, 35, 36, 61]. Because of these factors, this research studies 2-fingered end-effectors.

2.4 Grasp representation

To facilitate the automatic manipulation of a given object, it is essential to cast the problem in a way that captures manipulator-object interaction and promotes area-of-contact synthesis. A key component to problem formulation within this context, is grasp representation. Early works focused on representing grasps in terms of contact points for objects in 2-dimensional space [295, 296]. Chinellato, Fisher, Morales and Pobil for instance, proposed 10 evaluation criteria that assessed the suitability of 3-fingered grips for 2D parts [297]. The mathematical basis of their contact model is graphically depicted in Figure 2.15—a, b. Bicchi and Kumar review early contact models [298]. Later, Saxena *et al.* showed that a 3-dimensional grasping point representation could be used to facilitate a search algorithm based on local features from multiple viewpoints (Figure 2.15—c) [1, 59]. Le, Kamm, Kara and Ng questioned the usefulness of such a single point representation, noting issues related to grasp stability, grasp accuracy, computational efficiency and difficulties related to motion planning [87]. Instead, they framed a 2-fingered grasp location in terms of their pair of points representation, illustrated in Figure 2.15—d. Such pairs were selected by scoring gradient angle features using an SVM model [87].

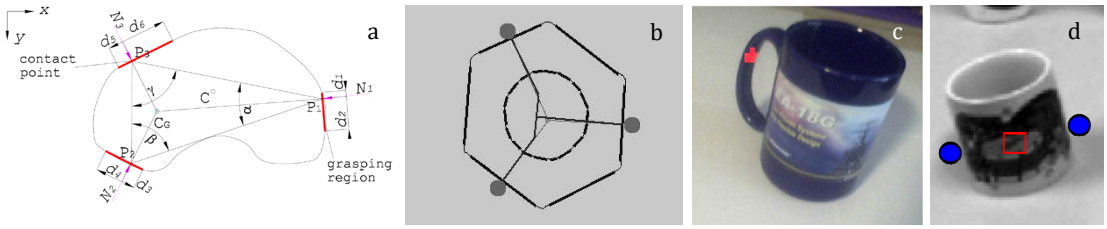


***Figure 2.15*. (a), (b)—the mathematical and geometrical basis of an early 3-fingered contact model [297]. (c)—the grasping point representation first proposed by Saxena *et al.* in 2007 [1, 59]. (d)—the pair of points representation proposed by Le, Kamm, Kara and Ng in 2010 [87].**

As learning-based methodologies progressed, some works noted the inherent sensitivity of the pair of points representation to appropriate feature definitions. Moreover, Jiang, Moseson and Saxena argued that this modality represents only part of the problem, leaving other aspects of the grasp to be estimated separately [65]. They addressed this issue with their oriented rectangular representation. A grasp pose was considered analogous to the sensory data directly related to the area of a parallel jaw gripper in the open position, approached from the perspective of the observer. Their representation is shown in Figure 2.16—a. The grasping rectangle representation proved highly congruent with data-driven, learning-based methodologies. By defining grasps in this way, deep learning techniques that define their own features could be used to generate and evaluate such locations—which they demonstrated in their later work using hand-labelled data and CNNs [13]. A plethora of works have since used analogous representations (Figure 2.16) [58, 68-71, 73]. Sun, Yu, Liu and Gu for instance, trained a neural network to classify grasping rectangles by extracting histogram of gradient features [66]. Pinto, Davidson and Gupta adapted the rectangular representation for 2-dimenstional vision, using basic intensity data [61]. In addition to the oriented grasping rectangle, Pinto and Gupta used a sample patch to provide their methodology with additional context related to the area surrounding the grasp [60]. Their representation is shown in Figure 2.16—e.

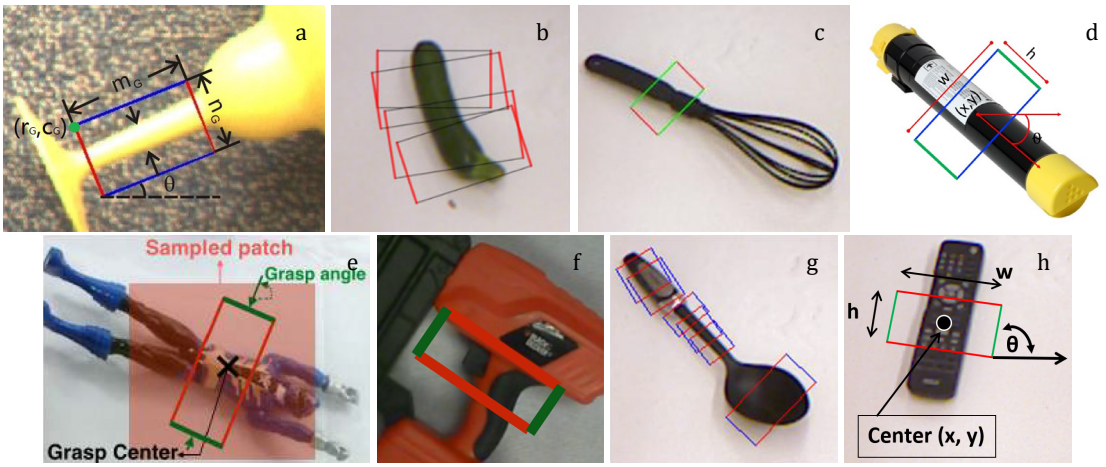


***Figure 2.16*. (a)—the oriented grasping rectangle representation first proposed by Jiang, Moseson and Saxena in 2011 for RGBD data [65]. (b), (c), (d), (e), (f), (g), (h)—various other representations similar to the oriented grasping rectangle [58, 60, 61, 66, 70, 71, 73].**

Various other grasp representations have also been proposed, e.g., an oriented bounding box [23], a Cartesian-based representation with orientation and width (Figure 2.17—a) [72], a gripper pose that incorporates grasp score (Figure 2.17—b) [42], feature points that relate back to robot configuration and motor torque (Figure 2.17—c) [77], contact points related to finger placement (Figure 2.17—d) [24], line-based representations (Figure

2.17—e) [35, 75, 76], a representation based on a query density function (Figure 2.17—f) [81], a heat map representation (Figure 2.17—g) [20], a 3-fingered region-based descriptor [25] and others (Figure 2.17—h) [40, 74].

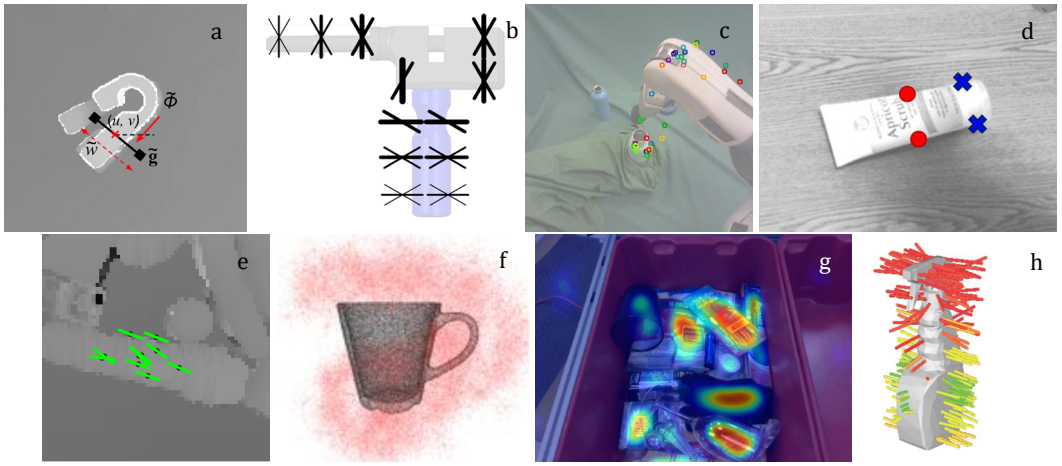


Figure 2.17. **(a)—Cartesian-based representation [72]. (b)—grasp pose representation related to gripper score function [42]. (c)—feature point representation [77]. (d)—grasp contact point-based representation [24]. (e)—line-based representation visualised in depth data [75]. (f)—visualisation of a query density function related to potential grasp areas [81]. (g)—heat map representation for suction-based end-effectors [20]. (h)—grasp representation based on predicted quality [74].**

Only a few representations other than the oriented grasping rectangle have managed to maintain widespread community interest, namely 3D pose estimation (Figure 2.18) [6-8, 19, 21, 62, 299-301]. This representation touches on concepts such as scene understanding and object awareness. By inferring the relative 6-DOF pose of objects within the scene of a robot, safe human collaboration and context-specific manipulation can be implemented. Generally, pose estimation is achieved via shape primitive estimation. Tremblay *et al.* for instance, trained a deep neural network to estimate the 6-DOF pose of known objects from a single RGB image for robotic manipulation (Figure 2.18—a) [14]. Their publication focused on bridging the *reality gap* by using a combination of real and synthetic data. Schwarz *et al.* used 6-DOF pose estimation in their 2017 APC entry [17]. Similarly, Zeng *et al.* predicted the 6-DOF pose of objects within a scene by fitting pre-scanned 3D object models [18]. Their methodology showed good, generalised object manipulation performance in the 2016 APC. Their pipeline output is shown in Figure 2.18—b. Recently, task-based representations have gained research interest [11, 302-304]. Manuelli, Gao, Florence and Tedrake for instance, proposed the keypoint affordance for category-level robotic manipulation (kPAM) framework [12]. In their work, they showed that purposeful manipulation could be achieved by representing an object in terms of geometric costs and constraints through semantic keypoints (Figure 2.18—g). Similarly, Qin *et al.* showed that deep neural networks could be used to learn keypoint representations for task-oriented tool manipulation [305].

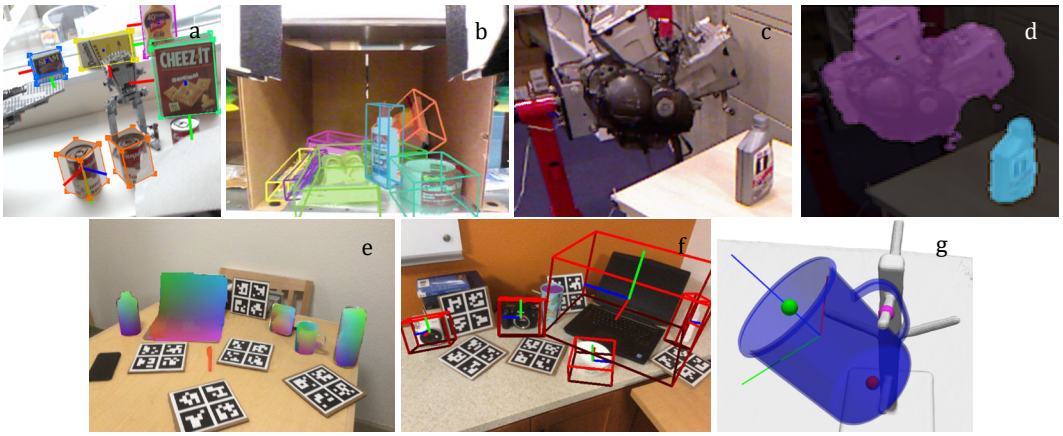


***Figure 2.18*. (a)—pose estimations of YCB objects using the DOPE method [14]. (b)—MIT-Princeton 6D object pose estimation during the 2016 APC [18]. (c), (d)—pose estimations using the SegICP method [300]. (e), (f)—pose estimation process examples using the NOCS method [299]. (g)—example of reference keypoints using the kPAM method [12].**

In this dissertation, a 2-dimensional variant of the oriented grasping rectangle utilising monocular RGB data is proposed. Though a wide range of representations have been studied, the oriented rectangle remains prevalent in contemporary research [58, 70, 73]. Furthermore, many works considered state-of-the-art utilise this representation. Pinto and Gupta for instance, demonstrated the utility of learning this representation for a self-supervised methodology [60]. Over the course of 700 robot hours of trial and error, their manipulator learned features within such rectangles that produced the desired outcome. Their later work implemented this representation for a sophisticated learning methodology based on game theory [61]. Pose estimation for robotic manipulation has shown promise. However, this technology is still in the early stages and therefore, imprecise. A recent paper published in 2019 in affiliation with Google for instance, reported a mean average precision of 65.1% for object instances where translational error was less than 10 cm and orientation error less than 10° [299]. They also reported a mean average precision of 40.9% for object instances where translational error was less than 5 cm and orientation error less than 5°. This degree of error makes grasping via pose estimation unsuitable for industrial application at this time. Moreover, the graphics card used, the NVIDIA TITAN Xp, is priced at approximately $2,000 USD [306], which may be prohibitively expensive for some researchers. Pose estimation is currently better suited to tasks where extended context is desired—as opposed to more subtle grasping tasks within industrial automation and manufacture. The grasping rectangle representation provides utility in that its shape and size can easily be adapted to the gripper and existing hardware. This representation also encourages the generation of many candidate grasps, which is central to the methodology proposed in this thesis.

2.5 Grasp synthesis methodologies

Often cited as one of the best-selling industrial robots of the 1980s, the Kawasaki EX100 was documented as achieving a positional repeatability of ±1.0 mm with a 100 kg payload and maximum arm speed of 90°/s [307]. A similar contemporary robot, the BX100S for example, has a documented positional repeatability of ±0.06 mm with a 100 kg payload and maximum arm speed of 155°/s [308]. With such longstanding hardware capabilities, it is increasingly evident that the related control theory is lacking. Intuitively human beings know how and where to grasp an object, regardless of prior knowledge. In this context,

robotic manipulation lags far behind human performance. For a broad comparison of animal and robotic manipulation, refer to *The Annual Review of Control, Robotics and Autonomous Systems* [309]. For a recent review of trends and challenges within generalised robotic manipulation, refer to the summary by Billard and Kragic [310]. Manipulation has been a central research topic within robotics and autonomous assembly for many years [311-314]. Despite the frequency with which this problem has been addressed, the task of automatically grasping and handling an object—with no *a priori* knowledge—remains challenging. This is well recognised throughout literature [9-13, 18, 19, 40-42, 50-55, 58-64, 73-75, 81, 82, 84, 85].

Given an object and gripper, there may be an infinite number of ways to interact with the object such that the grasping task is satisfied. Grasp synthesis refers to the methodology by which a suitable grasp configuration is selected from this large solution space. Broadly speaking, grasp synthesis methodologies can be grouped into two categories: analytic and empirical. Analytic methods are based on grasp formulations or models grounded on mathematical assumptions, such as contact models, friction and rigid body modelling [315, 316]. Analytic approaches dominated this field prior to the early 2000s [298, 311, 317, 318]. Hester, Cetin, Kapoor and Tesar for instance, proposed a criteria-based approach to grasp synthesis that utilised a grasp quality metric to approximate the amount of finger force required to resist external loads and maintain the grasp [319]. Similarly, Park and Starr found force-closure grasps for polygons based on a fingertip contact model and contact wrench assumptions [320]. Empirical—or data-driven—approaches derive models from experience. Earlier works focused on heuristics. Bowers and Lumia for example, proposed a rule-based expert system that generated candidate grasps via fuzzy logic [321]. Computational geometry was then used to gauge the quality of these potential grasps prior to selection.

Analytic approaches offer grasp assurances in the form of comprehensive mathematical descriptions of the interaction, provided a complete model of the object, gripper and system is known. Unfortunately, real-world systems are prone to sensory noise, making the acquisition of precise models difficult. Many works for instance, have noted the inaccuracy associated with RGBD sensors [72, 75, 76, 322]. Rodriguez, Cogswell, Koo and Behnke attempt to address this issue [90]. They proposed a method that infers an object model from incomplete and noisy RGBD data. Moreover, many systems are prone to kinematic inaccuracies and noise from the robot itself. Empirical approaches do not rely on complete knowledge of the system and instead, build rough models from experience, often requiring large amounts of data [35, 36, 75]. A disadvantage of empirical methodologies is that they can only be verified through testing.

ML has become the facilitator of choice for many modern data-driven grasp synthesis methodologies. Recent research publications see ML employed for automated pick-and-place assembly tasks within individualised manufacturing [323], bin-picking of small parcels for warehouse automation [324], grasp pose and orientation estimation [325, 326], ambidextrous grasping of novel objects from heaps [327], virtual grasp learning by failed demonstrations [328], object grasp affordance learning using deep residual U-nets [329] and unsupervised or semi-supervised learning for general novel object grasping [330, 331]. Kroemer, Niekum and Konidaris present a review of the role of ML in recent manipulation research [332]. The rise in use of ML for automated grasping is further supported by the trend in published articles that include keywords related to this topic. Figure 2.19 illustrates the number of search results from Google Scholar for various terms over time. Despite this

shift, non-ML-based grasp synthesis is still a somewhat active field of research. A recent publication by Sainul, Deb and Deb for instance, illustrated that suitable grasps could be found via a novel 3D object model slicing method [46]. Other methods have also been proposed, for example, Klingbeil *et al.* found grasps that maximised gripper contact area using depth information [24], Paolini, Rodriguez, Srinivasa and Mason proposed a data-driven statistical framework for post-grasp manipulation [322], Arruda, Wyatt and Kopicki investigated an approach based on object reconstruction from multiple view-points [78] and several works used basic colour and shape estimation [31, 57].

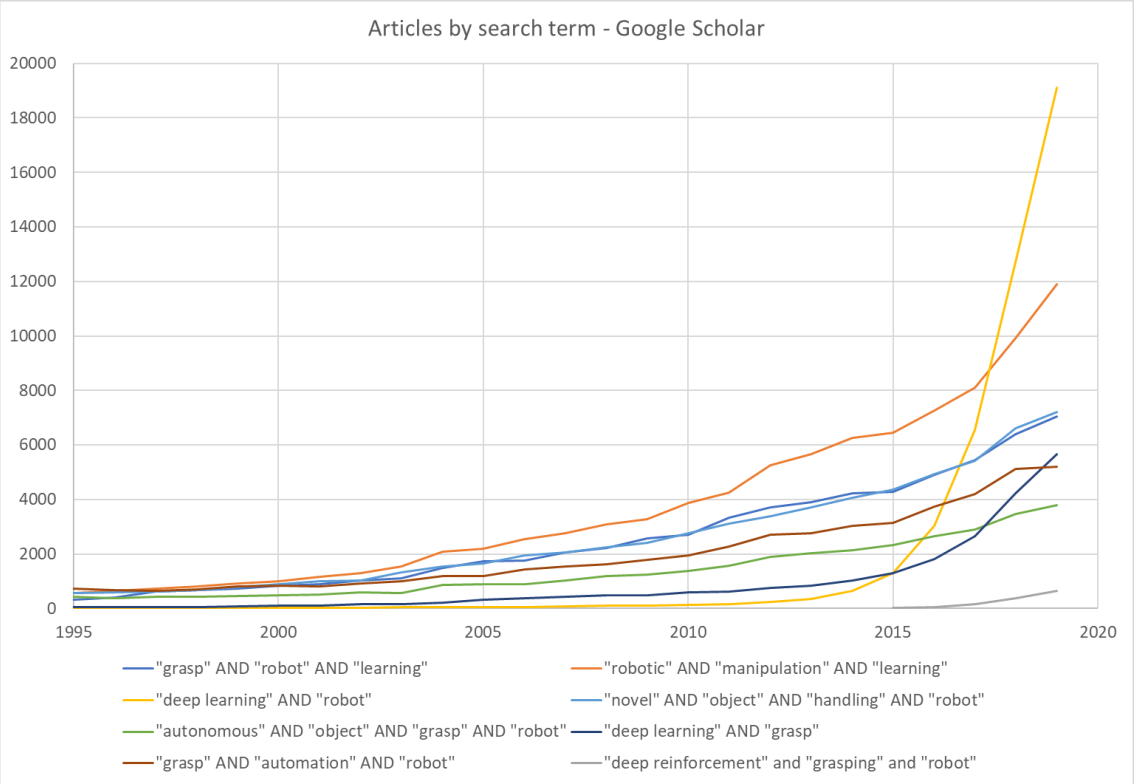


***Figure 2.19*. The number of returned articles by search term over time. Source: Google Scholar.**

In contemporary works, data-driven synthesis is commonly approached by first sampling a set of potential candidate grasps, followed by a process that ranks and selects based on some set of specified metrics. This form of synthesis is known as the generate-and-test methodology, popularised with ML by Lenz, Lee and Saxena in 2015 with their 2-step cascade methodology [13]. By sampling raw sensory data in terms of their oriented rectangular representation [65], they quickly generated a large set of initial grasp proposals. They achieved this by exhaustively evaluating potential areas of an image, each classified using a shallow CNN. A deeper network, with a larger number of hidden layers, then re-evaluated the top detections to facilitate a final selection from the candidate grasp pool. A graphical illustration of their methodology can be seen in Figure 2.20.

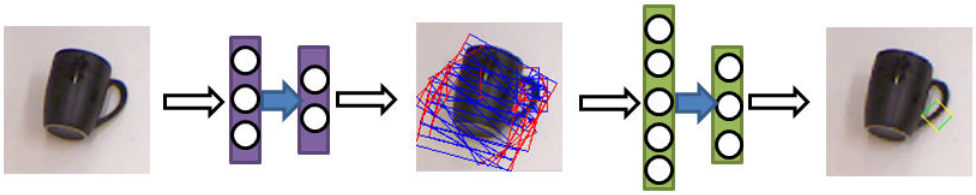


***Figure 2.20*. Illustration of the 2-step cascade detection process proposed for grasp synthesis by Lenz, Lee and Saxena [13].**

Numerous variations of this form of grasp synthesis have since been investigated. Sun, Yu, Liu and Gu for example, offloaded some network complexity to other non-learning processes [66]. By rotating their image prior to classification, they were able to simplify the task, thus simplifying their networks. They achieved this rotation by tabulating the response of a Sobel filter applied to their image [66]. Their process is illustrated by an analogous example in Figure 2.21. Redmon and Angelova further reduced computational overheads with their single-shot methodology (Figure 2.21—d) [69]. They opted for a much larger end-to-end network architecture that performed bounding box regression, thereby predicting candidate grasps from the entire image directly. Prediction time was reduced from 13.5 seconds—Lenz, Lee and Saxena [13]—to approximately 0.076 seconds. Watson, Hughes and Lida reproduced their work on a CPU machine, noting a grasp detection time of 1.8 seconds [106]. Cheng and Meng noted 0.14 seconds for a comparable network using GPU computation [68]. Chu, Xu, Patricio and Vela implemented a much larger bounding box regression network and noted an image to prediction time of 0.25 seconds [70]. Their network architecture consisted of 50 layers, compared to closely related works that used eight [69] or five [333]. Guo *et al.* added to the Redmon and Angelova methodology by incorporating tactile information in addition to visual sensing [333]. A plethora of other approaches have also paired the generate-and-test structure with some form of machine learning [10, 41, 42, 58, 60-62, 71, 73]. For a more in-depth review of data-driven grasp synthesis in which sampling and ranking is employed, the reader is referred to the review by Bohg, Morales, Asfour and Kragic [64].

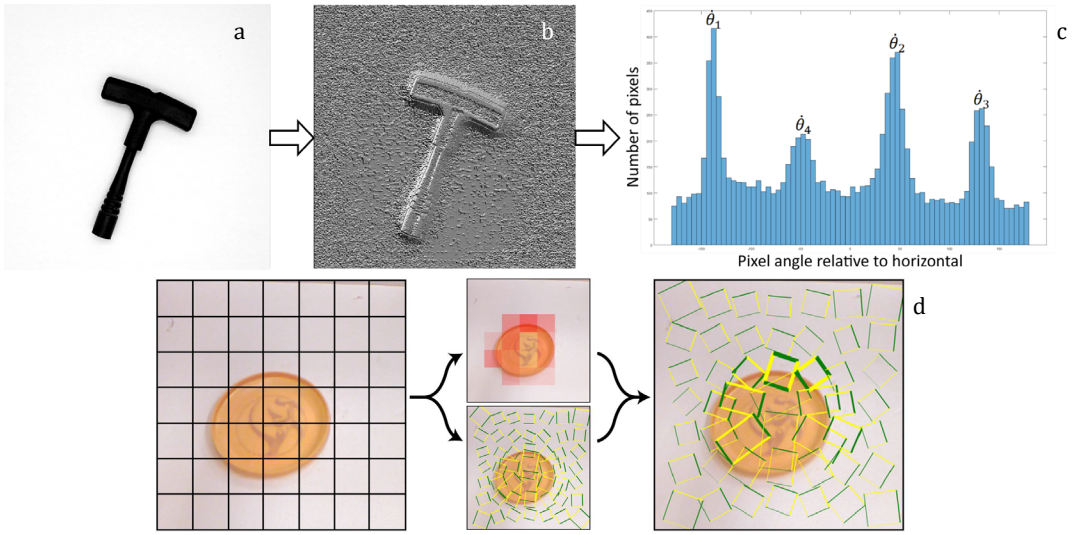


***Figure 2.21*. (a), (b), (c)—illustration of the process utilised by Sun, Yu, Liu and Gu [66] to rotate an image prior to classification. (a)—representation of a greyscale input image. (b)—representation of a Sobel magnitude version of the input image. (c)—statistics related to the Sobel response, used to rotate the input image accordingly. (d)—illustration of the MultiGrasp model proposed by Redmon and Angelova [69].**

Many generate-and-test methodologies produce their initial set of candidate grasps through supervised learning. To facilitate the candidate generation step of their methodology, Lenz Lee and Saxena created a dataset consisting of 1035 object images with manually labelled oriented grasping rectangles [13]. They gauged the effectiveness of their CNNs by comparing differences in rotation and overlap between ground-truth rectangles and rectangles generated by their networks. Figure 2.22—b illustrates some samples from the Cornell Grasp Detection Dataset [67], which is often used by other works for related purposes [40, 68-73]. Sun, Yu, Liu and Gu for instance, extract positive and negative training samples from real grasps recorded in the Cornell dataset [66]. Examples of patches corresponding to each respective class are shown in Figure 2.22—a. Kumra and Kanan also used this dataset for candidate generation [58]. Johns, Leutenegger and Davison used a physics engine to generate their 2,000-sample training dataset for grasp synthesis [42]. Trials that resulted in an object lift of over 20 cm were labelled as successful (Figure 2.22—c). Similarly, ten Pas, Gualtieri, Saenko and Platt generated their initial set of grasp candidates via a CNN trained for binary classification from partial object meshes [41]. An example of candidate force-closure grasps for their method is shown in Figure 2.22—d. Manuello, Gao, Florence and Tedrake trained their pipeline for task-specific keypoint recognition prior to task optimisation [12]. Their dataset consisted of 100,000 labelled samples of two object categories, annotated on 3D reconstructions of the objects. Figure 2.22—e, f shows an example of 3D keypoints detected by their network prior to action optimisation.

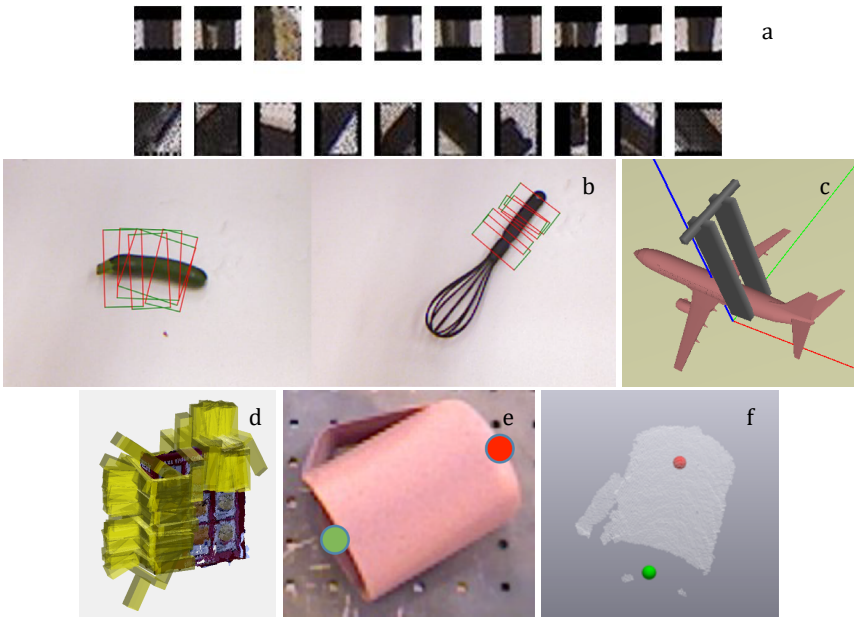


***Figure 2.22*. (a)—examples of patches extracted from the Cornell dataset [66]. Top images represent positive samples. Bottom images represent negative samples. (b)—samples from the Cornell Grasp Detection Dataset, with annotated rectangles shown [73]. (c)—simulated positive grasp sample [42]. (d)—candidate force-closure grasps [41]. (e), (f)—detected keypoints for the mug object in image format and depth reconstruction [12].**

More sophisticated methodologies toward automated grasping have also been proposed. Self-supervised learning for instance, has been well-studied. Pinto and Gupta demonstrated their approach which gradually learned to predict grasp locations via 50,000 self-supervised trial and error attempts [60]. Levine, Pastor, Krizhevsky and Quillen built on their work by collecting over 800,000 attempts over the course of two months, using between six and 14 manipulations at any given time [35]. Zeng *et al.* used self-supervision to learn complex relationships between pushing and grasping behaviours to isolate and

grasp the desired object from clutter [76]. Pharswan, Vohra, Kumar and Behera proposed a 5-stage, unsupervised methodology for grasping novel objects in clutter and isolation [330]. Deep reinforcement learning has also gained some traction in recent years, as evidenced by the trend in Figure 2.19. Neef, Luipers, Bollenbacher, Gebel and Richert proposed deep reinforcement learning for collaborative pick-and-place assembly tasks [323]. Strudel *et al.* used a 6-DOF UR5 manipulator to validate their reinforcement learning-based motion planner for everyday tasks, such as grasping or preparing a meal [303]. Their method learned to combine basic skills from only a few demonstrations in simulation. Figure 2.23—a, b illustrates their method transferring a skill from simulation to reality. Mudigonda, Agrawal, Deweese and Malik learned robust policies from 120,000 demonstrations that enabled the anthropomorphic grasping and lifting of objects within a simulated environment (Figure 2.23—c) [276]. Contrary to these works, Xie, Li, Zhang, Zhu and Zhu showed that virtual grasps could be learned via failed demonstration [328]. They employed model-free inverse reinforcement learning and a combination of successful and unsuccessful demonstrations to learn grasp policies in simulation. In each trial, an Oculus Rift VR controller was used to record the human demonstrator. Pinto, Davidson and Gupta demonstrated that robust grasping behaviours could be developed by training two reinforcement learners as adversaries [61]. The two opposing adversaries used in their method are shown in Figure 2.23—d. In their framework, one manipulator attempts to learn a grasp policy, while the adversary learns behaviours that result in grasp failure. Both policies are learned simultaneously. In doing so, each learner becomes better at their respective task. When the adversary is removed, the remaining model generates grasps that are more stable compared to grasps generated without an adversary. They showed a 14% improvement in grasp rate by employing an adversarial learner [61].

***Figure 2.23*. (a), (b)—skill learned via simulation, transferred to reality [303]. (c)—simulated anthropomorphic grasping [276]. (d)—adversarial learning framework [61]. The manipulator on the left attempts to learn the grasping task, while the right manipulator learns a snatching or grasp failure task.**

Closed-loop grasping is another sophisticated approach throughout literature. This technique generally employs some form of feedback to move or test an object prior to, or during a grasp to improve the grasp itself or ensure contact. One form of closed-loop grasping is visual-servoing, where an object or end-effector is moved using primarily visual feedback. Arruda, Wyatt and Kopicki illustrate this in their work [78]. They mounted a depth camera on a robot's wrist. Their vision system was driven to various locations autonomously to maximise object reconstruction, prior to making a grasp attempt. Sadeghi, Toshev, Jang and Levine trained a deep learning algorithm to determine which actions move their end-effector closer to the desired object, based on past movements [334]. Their work was viewpoint-invariant and robust to noise in calibration and measurement errors. Kalashnikov *et al.* employed a closed-loop methodology that continuously updated its grasp strategy based on recent observations [36]. Their self-supervised network leveraged over 580,000 grasp attempts to learn behaviours such as pre-grasp manipulation, grasp readjustment, object probing, grasp failure recovery and recovery from perturbations. Similarly, Marrison, Corke and Leitner demonstrated the capability of their 50Hz closed-

loop methodology in non-static environments where objects were subject to movement [72]. Many similar works have also applied visual-servoing techniques [9, 29, 71, 75, 335-337]. Pinto and Gupta employ force sensor feedback in their work to ensure suitable contact [60]. Ding, Paperno, Prakash and Behal use a combination of force and slip sensors to ensure object contact [80].

Autonomous grasping research is largely separated into two tasks: grasping objects in isolation and grasping objects in clutter. Each grasp modality has its advantages and disadvantages. Grasping in clutter is a recent development within this field—generally considered the more difficult task. Studied applications include clutter clearing tasks [10, 11, 27], home service robotics [8, 14, 75, 84, 327] and applications where objects need to be sorted into bins [35, 324]—tasks that generally do not require significant accuracy. ten Pas, Gualtieri, Saenko and Platt, studied the use of point clouds for grasp pose detection in clutter [41]. Figure 2.24—a illustrates an object being selected from clutter using this method. Harada *et al.* investigated bin picking for cylindrical objects (Figure 2.24—b) [56]. Pharswan, Vohra, Kumar and Behera proposed an unsupervised learning algorithm for the selection of feasible grasp regions within clutter [330]. Figure 2.24—c illustrates this method being used to grasp a tennis ball tube from a heap of miscellaneous objects. The APC was largely focused on grasping in clutter for order fulfilment and warehouse automation [16-18, 21]. The approach submitted by Zeng *et al.* for instance, showed generalised performance for grasping and recognised both novel and known objects in cluttered environments [20]. Their system successfully stowed a wide range of heaped objects from an unstructured tote into a structured storage system. Similarly, the method proposed by Schwarz *et al.* was required to pick specific objects from unordered shelf boxes [19].

***Figure 2.24*. (a)—example of cluttered grasping from an unstructured pile using point cloud grasp pose detection [41]. (b)—picking a cylindrical object from an unstructured pile [56]. (c)—picking a specific object from a heap of objects [330]. (d)—example of novel object grasping in dense clutter [62].**

Grasping in isolation can be defined as picking a single object in a scene where no two objects are touching, though there may be multiple objects present. Single-object grasping in isolation is usually more focused toward task-based manipulation [9, 12, 77], assembly [28, 121, 266, 292, 323, 338] or tasks where finer manipulation is desired. Guo for instance, show that higher quality single-object grasps can be achieved by combining deep learning with visual and tactile sensing (Figure 2.25—a) [333]. They assess the quality of their grasps through strain readings from each finger. Johns, Leutenegger and Davison assessed region-based quality for grasping single objects under gripper pose uncertainty [42]. Figure 2.25—b shows their methodology being used to grasp a toy gun. Similarly, Morrison, Corke and Leitner predict the quality of grasps at the pixel level (Figure 2.25—c) [72]. They showed good performance for a range of difficult objects and tested their method in cluttered environments. Many works that make use of the Cornell dataset, grasp objects in isolation. Chu, Xu and Vela provide one such example [70]. Their method can be seen grasping sunglasses in Figure 2.25—d.

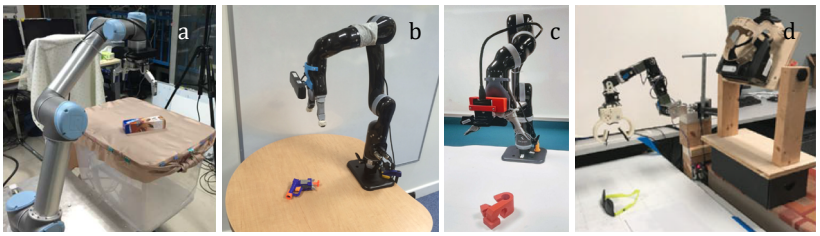


***Figure 2.25*. (a)—example of grasping in isolation [333]. (b)— example of deep learning used for grasping isolated objects under gripper pose uncertainty [42]. (c)—closed-loop grasping example that makes use of quality prediction [72]. (d)—experimental setup for methodology trained using the Cornell Grasp Detection Dataset [70]**

The research presented in this thesis is closely related to that of Lenz, Lee and Saxena [13], Sun, Yu, Liu and Gu [66] and other generate-and-test methodologies [42, 58, 73]. They exploit RGBD data and ML to generate numerous candidate grasps, of which a final grasp is selected. This research aims to improve on previous works by quantitively refining the candidate selection process. It is proposed to employ an RGB-based generate-and-test structure, paired with supervised learning techniques for grasping in isolation. These considerations are motivated by various issues with other approaches. Unsupervised learning for instance, requires extensive testing, i.e. not uncommonly in excess of 500,000 real-world samples [36]. Consequently, expensive equipment is needed that will survive these long test durations. Levine, Pastor, Krizhevsky and Quillen noted significant wear in the grippers of their manipulators, which collected over 800,000 grasp samples over two months [35]. This is problematic because unsupervised approaches capture these long-horizon degradations in their grasp policies. Moreover, acquiring appropriate hardware may be impractical. Supervised learning is more pragmatic in this respect. Pinto and Gupta for example, achieved a novel object grasp success rate of 66% from over 50,000 self-supervised grasp attempts [60]. In the same year, ten Pas and Platt demonstrated a novel object grasp rate of 88% from only 6,500 labelled samples [62].

Grasping in clutter is still a particularly difficult task, and therefore yields grasps that are not precise enough for industrial application. This is especially the case with many contemporary works, as RGBD sensors are often employed, which are well-known to produce noisy signals [72, 75, 76, 322] and struggle with precision lower than 30 mm [255]. Although research focused on isolated grasping has not yet produced a grasping methodology robust enough for industrial application, this form of grasping is well-established and therefore, is surrounded by a rich source of literature. In addition, assembly and many other manufacturing applications do not generally require the ability to grasp objects in clutter.

The generate-and-test structure provides versatility. Since the generation and evaluation stages are separate, processes that specialise in these tasks can be easily interchanged. This means that the entire framework becomes modular. This may be useful in cases where general performance is less important than specific performance. In past works, single-shot methodologies demonstrated increased temporal performance. This gain stemmed from slow memory read/write speeds between the GPU and other computer components. Larger networks take longer to evaluate and require a considerable amount of memory, but only need to be written into memory once. Although the networks used by generate-and-test-based methods are usually significantly smaller, they need to be processed many times—thus memory read/write time becomes an issue. The generate-and-test structure is still preferable for two reasons. First, the above-mentioned computational gain is not as significant with modern hardware. Second, overall memory requirements are generally

much lower for generate-and-test approaches, which enables implementation on light-weight hardware. Comparison with Redmon and Angelova illustrates the above two points. They demonstrated a single-shot performance of 13 FPS [69]. They used an NVIDIA Tesla K20. This GPU has 6 GB memory and is priced at US $5,000 [339]. In contrast, Kumra and Kanan achieved 16 FPS using a low-end workstation GPU and their generate-and-test structure [58]. They employed a Geforce GTX 645. This GPU has 1 GB memory and costs approximately US $70 [340].

2.6 Lack of standardisation

Reproducible benchmarks, protocols and metrics play a vital role in progressing a research field by providing quantifiable and comparable insight into the effectiveness of an approach. This is particularly evident in computer vision. Large-scale datasets such as Imagenet [104] and Microsoft COCO [105] for instance, advanced image recognition considerably by exposing gaps in the state-of-the-art—clarifying the direction for improvement. In manipulation research there is a current lack of general standardisation, which poses challenges in assessing and/or comparing various approaches throughout literature. This issue is well-established and long-standing [91-93]. For a general discussion related to replicable and measurable research within robotics, see the guest editorial by Bonsignorio and del Pobil [92].

Manipulation benchmarks are often considered from a task-based perspective. Triantafyllou *et al.* for example, propose an easy-to-reproduce benchmarking framework for pick-and-place systems in an industrial grocery setting [341]. They defined a testing protocol, set of evaluation metrics, setup description and an object test pool focused on fruits and vegetables. Cruciani *et al.* proposed an in-hand manipulation benchmark to evaluate how well a system can change the pose of an object while maintaining the grasp [342]. Yang, Lancaster, Srinivasa and Smith proposed a protocol that quantifies the manipulation and sequential manipulation skills of a two-armed robot with a standard 3x3 Rubik's cube [343]. Quispe, Amor and Christensen provide a comprehensive resource related to the taxonomy of benchmark manipulation tasks for service robots [344]. Over the past two decades, there has been increasing community interest in addressing the issue of repeatable research, with groups like the European Robotics Research Network (EURON) [345], the IEEE Robotics & Automation Society [346], DARPA [347], Amazon [15] and IROS [348] proposing a number of initiatives, workshops and competitions. The Amazon Picking Challenge for instance, has prompted improvement in the state-of-the-art for pick-and-stow tasks within unstructured, cluttered environments [16-22], centred around warehouse automation. However, none of these initiatives have adequately integrated all the subproblems relevant for benchmarking the wide range of grasping approaches.

Several self-contained platforms for benchmarking have been proposed throughout literature. Yang, Zhang, Pong, Levine and Jayaraman propose REPLAB—a reproducible combination of cost-effective hardware aimed to drive standardised experimentation across manipulation tasks [349]. Their platform consists of a robotic arm, RGBD camera and enclosed workspace, depicted in Figure 2.26—a. To further foster reproducible research, they define a grasp benchmark template, evaluation protocols, calibration details, performance measures, several baselines and a dataset of 92,000 grasp attempts. Their object test pool consisted of 20 *seen* objects and 20 *unseen* objects. Bottarel, Vezzani, Pattacini and Natale propose a similar approach toward benchmarking for

anthropomorphic manipulation (Figure 2.26—b) [350] and Murali *et al.* propose a benchmark for mobile robotics (Figure 2.26—c) [351].

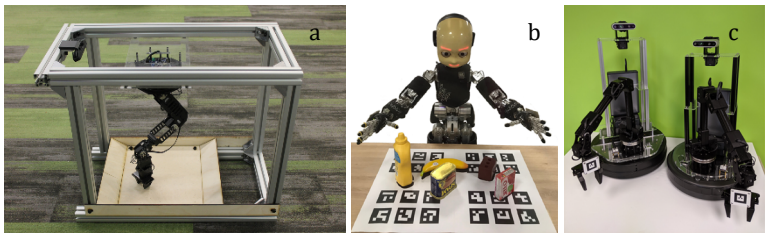


***Figure 2.26*. (a)—REPLAB benchmarking platform [349]. (b)—GRASPA 1.0 benchmarking platform for anthropomorphic manipulation [350]. (c)—PyRobot benchmarking framework for mobile robots [351].**

In many cases, authors throughout this field will cite the performance of their methodology in terms of grasp rate, which is typically defined as the rate at which their system was trialled to successfully grasp objects. To help facilitate this comparison, some have tried to define standardised object test sets, where the aim is to quantify performance from an object test perspective. The Yale-CMU-Berkeley (YCB) object and model set for instance, is a popular test set that consists of 75 objects from five classes: food items, kitchen items, tool items, shape items and task items [94, 95]. Each object is also accompanied by 600 RGBD scans and 600 high resolution RGB images. Examples of objects within their set are illustrated in Figure 2.27—a. Jamone, Bernardino and Santos-Victor utilised this object set in their work [352]. They defined a human-performance baseline by controlling an anthropomorphic robot manually. Similarly, Choi, Deyle and Kemp report a 43-object list intended to inform assistive robot design and benchmarking procedures, based on a survey of participants suffering from amyotrophic lateral sclerosis [353]. The Australian Research Council Centre of Excellence for Robotic Vision (ACRV) picking benchmark is another popular object test set [96, 97]. Their approach is focused on robotic challenges such as the APC. To help standardise their 42-object set, they provide an online list of the exact objects included in their benchmark. Links and shopping information are provided online in the form of a shared Google Spreadsheet [354]. Some of the objects used in their benchmark are shown in Figure 2.27—b. Pokorny *et al.* noted that object test sets that rely on shopping lists or pre-existing objects often suffer from a lack of worldwide availability [355]. To address this issue, they propose CapriDB—a database containing 40 3D printable mesh models of objects between 5-50 cm. By incorporating the use of 3D printers, a reproducible object set can be printed in a decentralised manner by anyone with the correct equipment around the world.

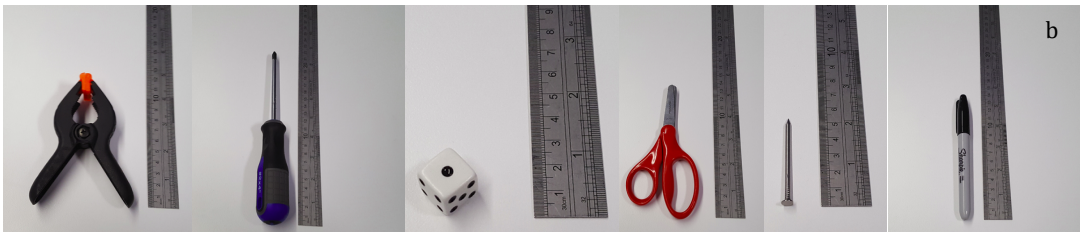


***Figure 2.27*. (a)—examples of objects from the YCB object and model set [94, 95]. (b)—examples of objects from the ACRV picking benchmark [96, 97].**

Trialling a physical system using a standardised set of objects can be time-consuming and relies on the availability of expensive hardware. Many works have sought to circumvent these requirements by proposing benchmark datasets—enabling rapid assessments of methodologies on real sensor data without any physical equipment. The Cornell Grasp Detection Dataset has been used extensively in this way [18, 20, 58, 68, 73]. Substantial effort has also been afforded to 3D mesh-model databases, in which models of real objects are captured and ported into simulation tools. The Columbia Grasp Database represents an established example in this respect [356]. Their dataset consists of 7,256 object models and 238,737 stable grasp annotations. BigBIRD is another example of an object instance database, containing 600 3D point clouds and 600 high resolution RGB images of 100 objects [357]. Georgakis, Reza, Mousavian, Le and Košecká repurposed part of their dataset in scene recognition for mobile service robots [358]. Several other mesh-model datasets have also been used by the robotics community, e.g., the KIT object model database [359], SHREC'14 Track [360], an RGBD dataset proposed for object and pose estimation for warehouse pick-and-place applications [361] and datasets intended for mobile manipulation in human environments [362]. GraspIt! [100, 363, 364] is a publicly available simulation environment commonly used in conjunction with such databases, though other similar tools also exist, e.g., OpenGRASP [99, 365], openRAVE [366] and VisGraB [101].

Gathering large amounts of real-world data is particularly costly for robotic grasping and can be avoided by leveraging simulation. Moreover, simulation allows researchers to establish and evaluate their approach prior to physical implementation. Unfortunately, it is unclear whether simulated environments currently resemble the real-world well enough to transfer learned behaviour. Though it is true that synthetic data can significantly increase the number of training samples, and simulated worlds make research timely and accessible, bridging the so-called *reality gap* has been problematic [367-370]. James *et al.* for instance, noted that their 99% simulated grasp rate decreased to 70% when tested in the real-world, using purely synthetic data [89]. However, further tuning utilising an additional 5,000 real-world grasp samples improved their grasp rate by 21%, quantified by 102 trials. A paper by NVIDIA also illustrates the *reality gap*. They generated photorealistic and domain randomised synthetic data for pose estimation [14]. Robotic experimentation using the YCB object set in clutter showed an 82% grasp rate from 60 trials.

Reproducibility is difficult in this field, as it is not always possible for researchers to experiment under the same conditions—nor is strict uniformity desirable, as researchers are working toward a wide range of applications, tasks and technologies. Many works consequently select for themselves a set of objects, tasks or objectives that they consider representative of the desired functionality. Unfortunately, this hinders any directly quantifiable comparison to other works and makes it difficult to assess relative performance.

Despite many efforts, standardised object test sets are not commonly used for comparison. Instead, works often use a *common household* or *common laboratory* object set—which varies and is self-defined, but is usually well-documented, e.g. Johns, Leurenegger and Davison use 20 everyday objects to empirically evaluate their methodology (Figure 2.28—a) [42], Mahler *et al.* use 10 novel household objects to test their approach (Figure 2.28—b) [40], Varley, Weisz, Weiss and Allen employ a dataset containing approximately 125 mesh models of everyday objects [85], ten Pas, Gualtieri, Saenko and Platt use 17 common household objects as their test set (Figure 2.28—c) [41], Zelenak *et al.* trial their approach using 39 common household objects (Figure 2.28—d) [53], Sainul, Deb and Deb use 24 household objects [46] and Goins, Carpenter, Wong and Balasubramanian use objects such as a cereal box, remote, glass and can (Figure 2.28—e) [82]. Some have suggested that the lack of community adoption of standardised object test sets stems—in part—from a lack of standardisation of sensing and manipulation hardware [64][72].

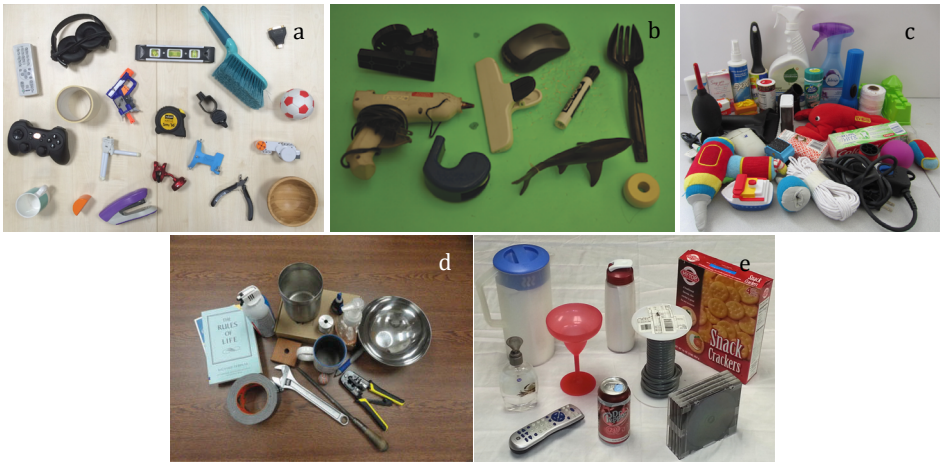


***Figure 2.28*. (a)—the 20 objects used by Johns, Leurenegger and Davison to evaluate their methodology [42]. (b)—novel objects used by Mahler *et al.* [40]. (c)—common household objects used by ten Pas, Gualtieri, Saenko and Platt [41]. (d)—sample of the training set used by Zelenak *et al.* [53]. (e)—objects used for grasp generation by Goins, Carpenter, Wong and Balasubramanian [82].**

Some papers assume the validity of annotated grasps in benchmark datasets. Kumra and Kanan [58], Wang, Li, Wang and Liu [71] and Redmon and Angelova [69] for example, exclusively train and test their approach using the Cornell Grasp Detection Dataset. Although this approach facilitates an objective comparison between methodologies, it is based on the assumption that the annotated grasps accurately map to real-world grasps and that hardware with similar functionality is used. In such cases, dataset accuracy is often interpreted as a potential grasp rate in practice. Evidently, datasets suffer from the *reality gap*—quantified by the drop in performance often reported by works through real-world validation. Mahler *et al.* for example, show a 20% disparity between dataset performance and real-world performance when grasping unknown objects [40]. Pinto and Gupta note dataset to real-world grasp rate disparities of 23% and 11% for known and unknown objects, respectively [60]. ten Pas and Platt note a nine percent disparity [62] and Chu, Xu and Vela note a seven percent disparity [70]. This disconnect may be due to the limited sample space that such datasets represent.

To help mitigate comparison issues, works throughout this field typically cite their real-world performance in binary terms, where a trial is framed as either successful or unsuccessful, i.e. $pass$ or $fail$. Unfortunately, this mode of comparison is problematic because what constitutes a successful grasp is not well-established and therefore, tends to

be defined by the author. Ding, Paperno, Prakash and Behal consider an initial grasp attempt successful if the gripper force sensors register a measurement in excess of 0.5 N and do not detect a slip velocity over 0.1 mm/s [80]. Several other works also utilise force sensors when defining a successful grasp outcome [45, 54, 79]. Many works pose task-based definitions, e.g., object transportation to a bin [24, 36, 53, 62], complex task descriptions [9, 12], shake protocol [61, 82] and lifting the object above some pre-defined height [1, 16, 42, 51, 52, 59, 60, 65, 72, 81]. Based on these loose definitions, the current state-of-the-art grasp rate sits between 85-95%. Table 2.1 provides more details.

***Table 2.1*. Cited grasping success rates of various 2-fingered, gripper-based works, where objects were physically trialled in isolation. Number of trials per object, methodological approach and testing conditions varied. Reproduction of Table 1.1.**

Year	Authors	Approach	Grasping success rate (%)	
			Known objects	Unknown objects
2007, 2008	Saxena *at al.* [1, 59]	2+ RGB. Logistic regression	90.0	87.8
2008	Saxena *et al.* [86]	RGBD. Probabilistic model	-	76.0
2010	Le *et al.* [87]	RGBD. Supervised learning	-	81.9
2011	Klingbeil *et al.* [24]	RGBD. Grasping function	-	91.6
2011	Jiang *et al.* [65]	RGBD. Supervised learning	-	87.9
2014	Kopicki *et al.* [81]	RGBD. Supervised learning	88.0	86.0
2015	Sun *et al.* [66]	RGBD. Supervised learning	86.0	86.0
2015	ten Pas *et al.* [62]	RGBD. Supervised learning	-	88.0
2016	Pinto *et al.* [61]	RGB. Adversarial learning	82.0	82.0
2016	Pinto *et al.* [60]	RGB. Unsupervised learning	73.0	66.0
2016	Johns *et al.* [42]	RGBD. Supervised learning	-	80.3
2017	Mahler *et al.* [40]	RGBD. Supervised learning	93.0	80.0
2017	Viereck *et al.* [75]	RGBD. Supervised learning	-	97.5
2018	Lévesque *et al.* [88]	RGBD. Scooping grasp	-	84.1
2018	Chu *et al.* [70]	RGBD. Supervised learning	-	89.0
2018	Kalashnikov *et al.* [36]	RGB. Reinforcement learning	-	96.0
2019	James *et al.* [89]	RGB. Reinforcement learning	-	91.0
2019	Manuelli *et al.* [12]	RGBD. Reinforcement learning	-	91.7 (mugs) 98.3 (shoes)

Though defining a grasp as either a $pass$ or $fail$ has been a somewhat pragmatic standard for comparison, training and assessment purposes, it is unclear whether this single-faceted measurement is related to grasp quality, handling quality or placement quality. Many works demonstrate this by achieving good grasp rates while clearly displacing the object, thereby compromising quality, e.g., Figure 2.29—a illustrates rotational error introduced by a grasp [35], Saxena, Driemeyer, Kearns and Ng illustrate droop when grasping a book without consideration of the centre of mass of the object (Figure 29—b) [59], Mahler *et al.* demonstrate error introduced by the gripper due to gripper-centre offset (Figure 29—c) [40] and Figure 29—d, e, f illustrates predicted grasp rectangles that may introduce some error when grasping the object when implemented. The work in this thesis is concerned with grasp outcome as well as grasp quality. As such, an object-agnostic, system-invariant methodology is proposed that objectively evaluates a grasp trial and aims to provide more descriptive power than the traditional $pass/fail$ notion of grasp outcome. In addition, the proposed measuring techniques may be of interest to other similar works as a basis for comparison and assessment. Implementation is simple, does not depend on the gripper or robot and only requires a single top-down camera.

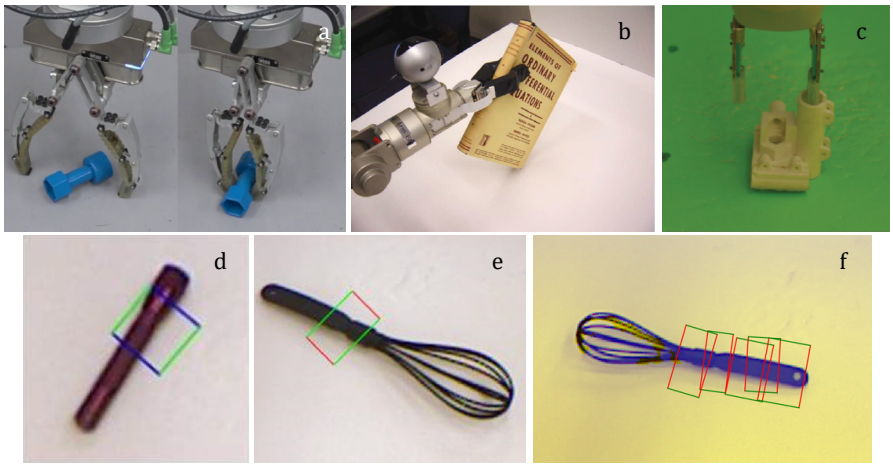


***Figure 2.29*. (a)—illustration of rotational error introduced by grasp attempt [35]. (b)—illustration of droop when grasping a heavy object [59]. (c)—illustration of translation-based error introduced by a grasp [40]. (d), (e), (f)—predicted grasp rectangles that may result in significant error when implemented [66][71][73].**

Even though grasp rate has become a commonly cited criterion that provides some comparable measure of performance, many works throughout literature inappropriately test their approach. Typically, the results cited by such works are not statistically significant due to an insufficient number of experimental samples. Morrison, Corke and Leitner for instance, cite a 100% novel object grasp rate in isolation, but only attempt four grasps per 10 objects, for a total of 40 trials [72]. Assuming simple random sampling, this gives a maximum margin of error at 95% confidence of ± 15.5%. Brown *et al.* also achieved a 100% grasp rate for their granular jamming gripper from a total of approximately 150 trials [55]—yielding a maximum margin of error at 95% confidence of roughly ± 8.0%. Many other works do not comprehensively test their approach either, e.g., Jiang, Moseson and Saxena achieved a 87.9% grasp rate for known and unknown objects from 120 trials [65], Gualtieri, ten Pas and Platt reported a 99% grasp rate for bottles and mugs from 112 trials [11], Kalashnikov *et al.* noted a 96% grasp rate in clutter from 102 trials [36], Chu, Xu and Vela noted a 89% single object in isolation grasp rate from 100 trials [70] and Klingbeil *et al.* reported a single novel object in isolation grasp rate of 91.6% from 120 trials [24]. In many cases, the number of trials used to test a methodology within this area of research is equal or less than 150 [14, 27, 35, 40, 61, 62, 78]. This dissertation aims to develop and implement the proposed methodology for proof-of-concept. The prototype system is to be trialled extensively, such that the results are statistically relevant. Two rounds of experimentation are to be completed. For comparison purposes, the literature-based definition of a successful grasp attempt is taken as a physically performed grasp in which an object is lifted vertically to a height of 15 cm and then held stationary for at least 10 seconds without falling. The object must be solely supported by the gripper, so that no other part of the object touches any other hardware.

2.7 Methodology implementation

Python (Figure 2.30—a) is a far-reaching programming language that has been widely used for industry and research. The Large Hadron Collider for instance, managed data using Python [371]. Because Python is open-source, a wide range of community libraries have been cultivated by collaboration, i.e. in March 2020, Python had over 220,000 modules available for download [372]. TensorFlow (Figure 2.30—b) is a very popular open-source Python library, originally developed by the Google Brain team to perform a range of tasks

related to AI, namely ML and deep neural networks [176, 373]. TensorFlow and Python are commonly used throughout this field. Cheng and Meng for instance, used TensorFlow for their CNN-based, grasp pose detection approach [68]. Similarly, Caldera, Rassau and Chai implemented their generate-and-test methodology on a Linux-based machine, using TensorFlow for their deep learning component [73].

***Figure 2.30*. (a)—Python logo. Source: *python.org*. (b)—TensorFlow logo. Source: *tensorflow.org*. (c)—MathWorks logo. Source: *mathworks.com*.**

Python offers two major tools for application within robotic grasping. First, scikit-image is a prolific and comprehensive image processing library [374]. Second, TensorFlow provides a versatile and ultra-customisable platform for constructing and training ML algorithms. Unfortunately, Tensorflow is notoriously difficult and time-consuming to install and has known stability issues on Windows-based machines—which has led many researchers to use other programming languages. Matlab for instance (Figure 2.30—c), is often employed throughout this field [34, 47, 53, 107]. The group that proposed the highly-influential grasping rectangle for example, used Matlab to implement and trial their approach [13]. Consequently, the Cornell Grasp Detection Dataset is formatted for Matlab interpretation [67]. Sainul, Deb and Deb use Matlab for their object-slicing-based grasp planner [46]. Matlab encourages low implementation time by simplifying their ML and vision tools, reducing turnaround times—which is ideal for quickly establishing a methodology within an initial research context. Currently, TensorFlow and Matlab only support NVIDA GPUs. It should be noted that Matlab is not as optimised or customisable as Python and is not open-source. Due to the above reasons, this dissertation prefers Matlab as an initial programming environment for this stage of research.

2.8 Literature conclusion

Automated robotic manipulation and peripheral topics have been covered in this chapter. Despite the many ways in which this topic has been approached, the automatic grasping of previously unseen objects by a robotic system remains a challenging task, for which there is no robust solution. Earlier approaches centred on gripper design. As sensing technologies improved, 2-fingered gripper implementations boomed in popularity, shifting the emphasis to control methodology. Works throughout this field often focus on grasping within a household setting, but industry-based works are not uncommon. AI techniques are ubiquitous in contemporary manipulation research and have significantly improved the state-of-the-art. ML, in particular, has been used very successfully when paired with a generate-and-test structure. Standardisation is a current issue within this field—with many works citing their grasp rate as one metric for comparison. Most works focus on grasp rate, which has left grasp quality lacking. Moreover, grasp rate is not consistently defined, nor is it validated with the same set of objects, confounding reproducibility. Additionally, grasp rate in and of itself cannot be used to assess grasp quality, a factor that is crucial for industry implementation. In this respect, an object-agnostic, system-invariant measurement that objectively quantifies system performance is missing.

Chapter 3

Grasp-induced error analysis

3.1 Grasp-induced error

Grasp-induced error can be defined as a change in object pose introduced by the robotic picking action itself. The resultant error can be observed by comparing the pose of an object pre-grasp with the pose of the object post-grasp, assuming the object has been picked and placed at the same location. Assessing the change in object pose in this manner accounts for any error introduced by the interaction between the gripping action and the object, positional error related to the robot and errors in vision. Grasp-induced error can be singled-out by minimising—and accounting for—errors in vision and hardware.

Grasp-induced error affects the quality with which objects are grasped, manipulated and subsequently placed. Compensating for this error is particularly important for open-loop methodologies that have no way of measuring this error once the gripper contacts the object. Although closed-loop feedback has been considered as a potential solution by some works, attaining an accurate object pose within the gripper during the grasp is impractical. As such, works usually employ closed-loop control from a grasp rate perspective, sensing and re-grasping an object if it has fallen out of the gripper. Kalashnikov *et al.* demonstrate this with their approach [36]. Figure 3.1 illustrates a re-grasp strategy learned by their methodology, in which an object grasp is re-attempted after initially failing to grasp. Several works also apply closed-loop control to continually re-evaluate and improve their selected grasp as the gripper approaches the object [71, 72, 75]. This dissertation investigates whether the quality of grasps can be improved by minimising grasp-induced error, and whether higher quality grasps lead to improved grasp rates. Also, by selecting grasps that promote low translational error, the pose of an object within the gripper can be more accurately known—which is key for many applications, particularly within industry. Though this method of measurement applies to any end-effector, this research is specifically concerned with the grasp-induced error associated with 2-fingered grippers as such configurations serve the research purpose while simplifying the gripper structure and subsequent implementation.

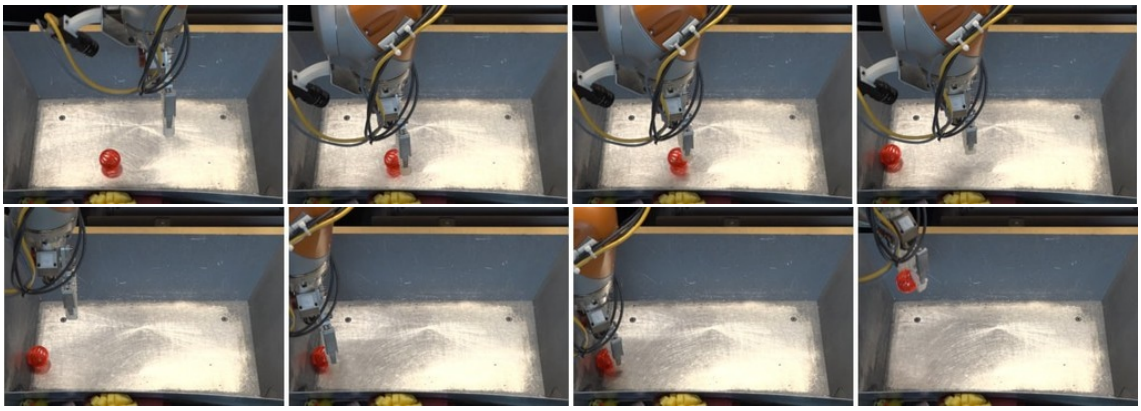


Figure 3.1. **Re-grasp strategy learned by the QT-Opt approach. This series of images illustrates an initially failed grasp attempt in which the object is perturbated. Their approach senses the failed grasp and re-attempts, successfully grasping the object in the second try. Source: Kalashnikov *et al.* [36].**

For 2-fingered grippers, grasp-induced error typically stems from several sources. Generally, a grasp that is well-aligned with the contact area will result in little or no measurable difference between pre-grasp object pose and post-grasp object pose. Such grasps tend to be stable. A grasp is considered stable if there is no gripper-relative movement of the object after the two fingers of the gripper are in their closed position. Figure 3.2 conceptually illustrates the result of a well-aligned grasp, accompanied by the real-world example in Figure 3.3.

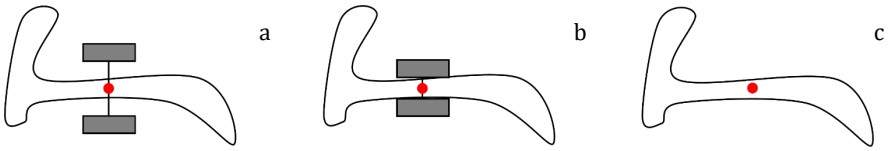


Figure 3.2. **Illustration of minimal grasp-induced error from a top-down view. (a)—open-gripper placement and object pose prior to grasp. (b)—closed-gripper placement and object pose after the grasp. (c)—resultant change in object pose due to the grasping action.**

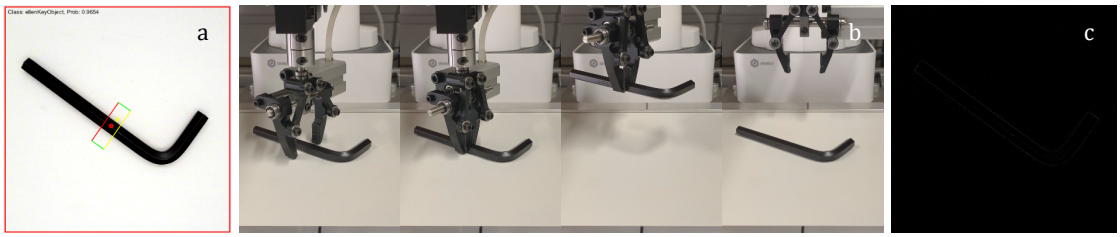


Figure 3.3. **Example of a well-aligned grasp. (a)—grasp selected for implementation. (b)—sequence of images illustrating the grasp trial. (c)—resultant difference image between pre-grasp image and post-grasp image. Note that this image appears blank as there was negligible error introduced by this grasp pose.**

In the case that the gripping surface of the object is well-suited to the two parallel plates of the gripper, but the approach is incorrect, two modes of error are commonly observed. Namely, error cause by a translational offset and error caused by a rotational offset. Translational error is induced by an offset between the centre of the gripper and the centre of the object within the purview of the gripping interface. Provided the end-effector is not forced to move with the grasp, the object will be translated such that it settles to the centre of the gripper in its closed position. Generally, such grasps are stable and induce a static error. Induced translational error is illustrated in Figure 3.4 and Figure 3.5.

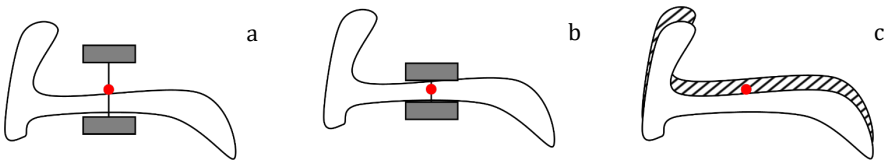


Figure 3.4. **Illustration of grasp-induced error from a top-down view due to gripper offset. (a)—open-gripper placement and object pose prior to grasp. (b)—closed-gripper placement and object pose after the grasp. (c)—resultant change in object pose due to the grasping action.**

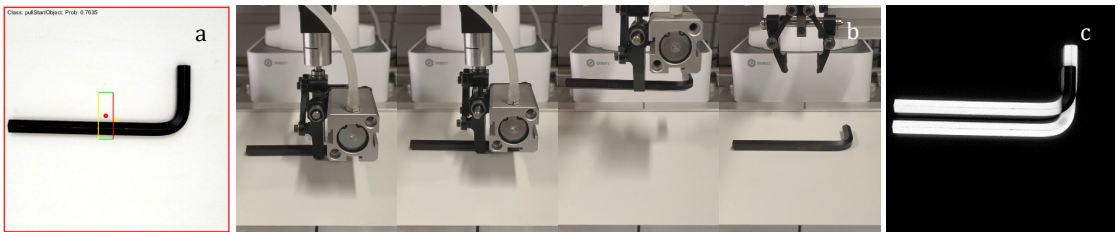


Figure 3.5. **Example of translational error. (a)—grasp selected for implementation. (b)—sequence of images illustrating the grasp trial. (c)—resultant difference image between pre-grasp image and post-grasp image.**

Rotational error is introduced when the two plates of the gripper are not parallel with the gripping interface—forcing the object into alignment, depicted in Figure 3.6 and exemplified in Figure 3.7.

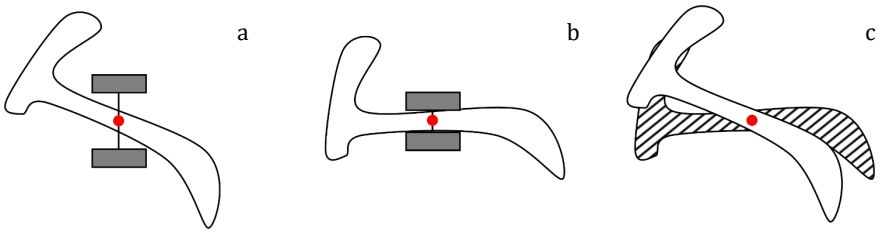


Figure 3.6. **Illustration of grasp-induced error from a top-down view due to a poorly oriented gripper. (a)—open-gripper placement and object pose prior to grasp. (b)—closed-gripper placement and object pose after grasp. (c)—resultant change in object pose due to the grasping action.**

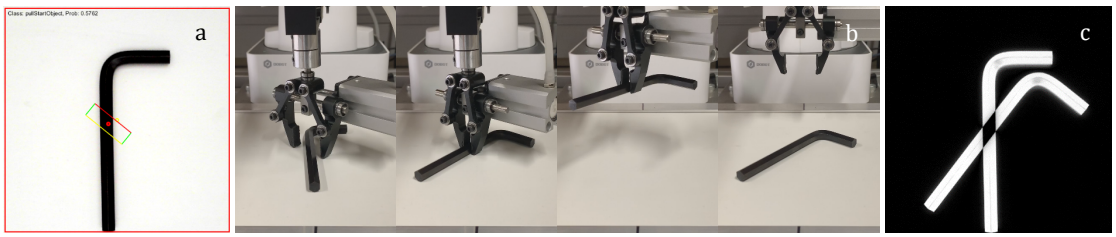


Figure 3.7. **Example of rotational error. (a)—grasp selected for implementation. (b)—sequence of images illustrating the grasp trial. (c)—resultant difference image between pre-grasp image and post-grasp image.**

Geometry-induced error is more difficult to model and can result in both stable and unstable grasps. In cases where the two plates of the gripper are not geometrically suited to the local gripping area, the object may be forced into some gripper-relative position. The outcome of these types of grasps is often unstable, causing objects to shift or dislodge peri-grasp. Open-loop methodologies cannot compensate for this dynamic error. However, the result can be measured post-grasp. By noting the grasp area and corresponding outcome, relationships can be formed that predict grasp outcome based on pre-grasp area. Displacement resulting from a poor gripper-object interface is illustrated in Figure 3.8 and Figure 3.9.

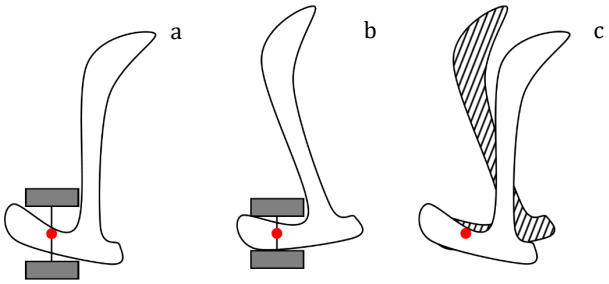


Figure 3.8. **Illustration of grasp-induced error from a top-down view due to poor gripper-object interface. (a)—open-gripper placement and object pose prior to grasp. (b)—closed-gripper placement and object pose after the grasp. (c)—resultant change in object pose due to the grasping action.**

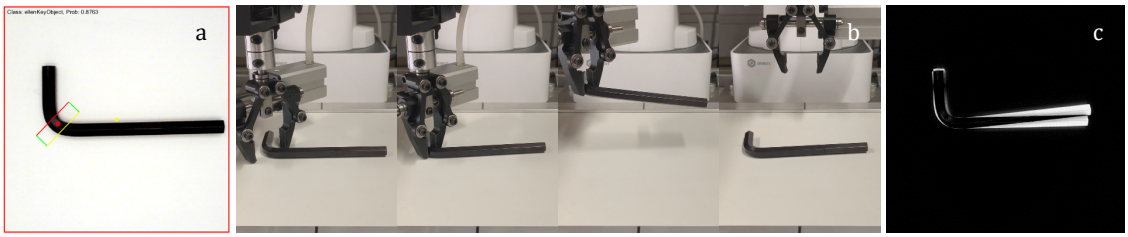


Figure 3.9. **Example of grasp-induced error due to poor gripper interface. (a)—grasp selected for implementation. (b)—sequence of images illustrating the grasp trial. (c)—resultant difference image between pre-grasp image and post-grasp image.**

An additional dynamic error often arises when the centre of gravity (COG) of the object to be grasped is not considered. When the object is heavy and picked at a location sufficiently offset to the COG, a torque is induced about the grip centre location due to gravity, provided the gripping force is insufficient to overcome this torque. Generally, the object droops, which results in poor manipulation quality and usually some small translational error. Figure 3.10 and Figure 3.11 illustrate a typical grasp outcome for a heavier object that lacks consideration of its COG.

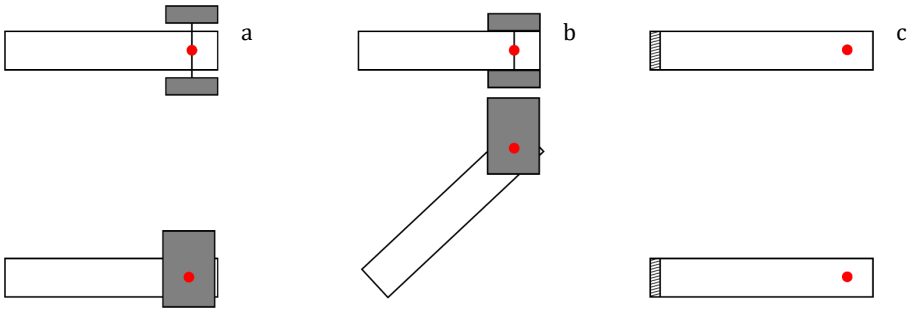


Figure 3.10. **Illustration of grasp-induced error from top-down and side views when grasping a heavy object away from its centre of gravity. (a)—open-gripper placement and object pose prior to the grasp. (b)—closed-gripper placement and object pose after grasp. (c)—resultant change in object pose due to the grasping action.**

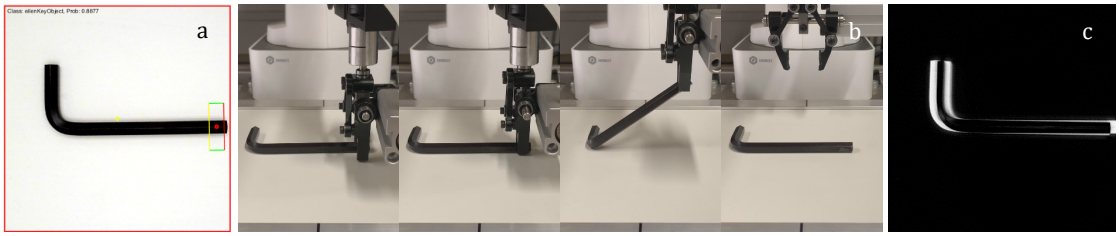


Figure 3.11. **Example of gripping a heavy object without consideration of its COG. (a)—grasp selected for implementation. (b)—sequence of images illustrating the grasp trial. (c)—resultant difference image between pre-grasp image and post-grasp image.**

Many other static and dynamic sources of error that have not been mentioned here also exist, e.g., friction, object deformability, etc. In practice, grasp-induced error is expressed as a combination of all the above-mentioned sources of error. However, a single source of error is typically most prominent.

3.2 Similarity metrics

Upon proposing the oriented grasping rectangle, Jiang, Moseson and Saxena also proposed new criteria to quantify the fit of their CNNs [65]. They compared the overlap and change in orientation between rectangles predicted by their CNN to ground-truth rectangles hand-annotated in their dataset. A grasp was considered correctly predicted if the orientation error was less than 30° and overlap was more than 25% [13]—providing an objective standard to assess and improve the performance of their neural networks with respect to their dataset. Many works used this methodology to improve the predictive power of their own networks by reducing the allowable error. Wang, Li, Wang and Liu for instance, increased the overlap threshold to 30% [71]. Moreover, this classification has been used very successfully as a baseline for comparison for works that employ the Cornell Grasp Detection Dataset [58, 66, 69, 70, 73].

Jiang, Moseson and Saxena quantify the quality of their predicted rectangles by computing rotation and overlap. Thresholds are then applied to frame grasp outcome in terms of $pass/fail$ or successful/unsuccessful. In this thesis, similar metrics are exploited to quantify grasp outcome quality—as opposed to rectangle recognition performance. Moreover, binary thresholding is avoided to yield continuous scores to provide a spectrum of grasp outcome. This work measures grasp-induced error through vision from a single top-down perspective. First, an image of the object is captured. The object is then picked, lifted and subsequently placed at the same location. By solely supporting the object with the gripper, weight is taken off the surface to remove any friction which may interfere with the measurement. A final image is then captured of the result. By comparing these two images, error introduced by the gripping process itself can be observed, which is a key factor for improving grasp, manipulation and placement quality of candidate grasps generated by a CNN.

Within this dissertation, two criteria referred to as *similarity metrics* are computed: OS and OE. The overlap score OS is described by the Jaccard similarity index—introduced by Paul Jaccard in 1901 [375]. This index is commonly used for object detection tasks that employ ML [376, 377]. For a more comprehensive description, see the DeepAI blog post on the topic by A. Rosebrock [378]. OS is computed as the intersection over union (IoU):

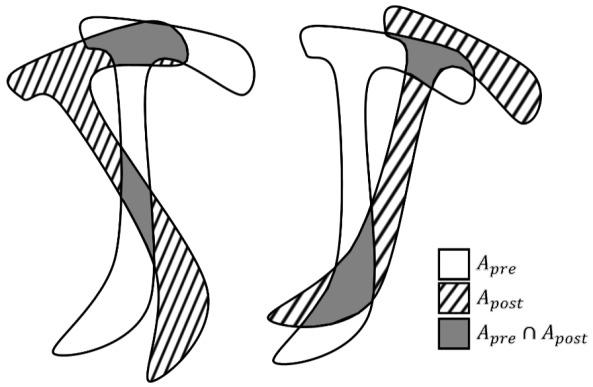


Figure 3.12. **Illustration of pre-grasp object area (A_{pre}), post-grasp object area (A_{post}) and the area of intersection between pre-grasp and post-grasp images ($A_{pre} \cap A_{post}$). Two examples of grasp outcome are shown.**

$$OS = \frac{|A_{pre} \cap A_{post}|}{|A_{pre}| + |A_{post}| - |A_{pre} \cap A_{post}|} = \frac{|A_{pre} \cap A_{post}|}{|A_{pre} \cup A_{post}|} \tag{3.1}$$

where A_{pre} is the top-down area of the object prior to the grasp and A_{post} is the area of the object after the grasp. Figure 3.12 illustrates the area of intersection between pre-grasp and post-grasp images used to calculate OS in Equation 3.1. This criterion measures overlap, strongly responding to translation error, though rotational misalignment is also captured. OS ranges from $[0, 1]$. As the area of overlap tends toward 0%, this score also tends toward 0. Conversely, OS will register a value of 1 in the case of a perfect grasp, where no difference between images is measured. The orientation error score OE compares the rotational change of the object:

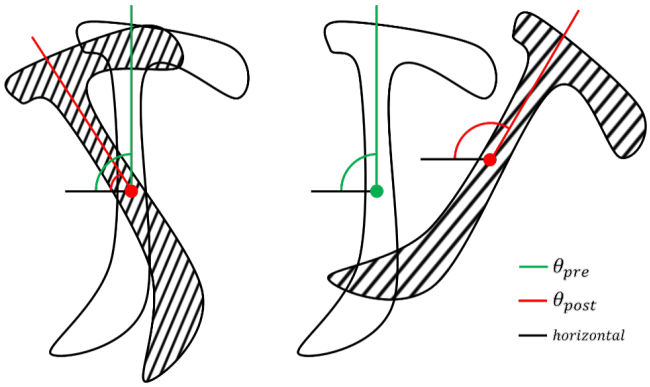


Figure 3.13. **Illustration of pre-grasp object angle (θ_{pre}) and post-grasp object angle (θ_{post}), relative to the horizontal axis of the image ($horizontal$). Two examples of grasp outcome are shown.**

$$OE = 1 - \frac{|(\theta_{post} - \theta_{pre})|}{180} \tag{3.2}$$

where θ_{pre} is the major orientation of the object pre-grasp and θ_{post} is the major orientation of the object post-grasp. θ_{pre} and θ_{post} are measured between the horizontal axis of the image and the major axis of the object before and after a grasp attempt, respectively, and range from $[0°, 180°]$. This convention was selected as per the default Matlab *regionprops* function output. Note that a range of $[0°, 360°]$ would also be appropriate, provided the denominator is adjusted accordingly. This metric scores the orientational change of an object introduced by the executed pick-and-place action. It does not respond to

translational error—as illustrated in Figure 3.13. OE ranges from $[0, 1]$. The relationship between this score and the angular difference between pre- and post-images is linear. As the angular change tends toward 180°, OE tends linearly toward 0. If no orientation change is measured, OE produces a value of 1.

The proposed metrics may be of interest to other similar works for improvement and as an objective baseline for comparison between methodologies, since they are not dependent on the system or object test pool. OS and OE measurement is relatively simple, requiring only a single top-down RGB camera and basic DIP processing. Since OS and OE are continuous, they can be used to train regression models and assess grasp quality. Alternatively, classification models can be trained via class-based thresholds. Although it is the intention of this work to define grasp quality and outcome in terms of OS and OE, the binary $pass/fail$ metric is retained for comparison and assessment purposes. Refer to Chapter 9.3.6 for implementation details related to OS and OE.

Chapter 4

3-Stage novel object handling methodology

4.1 3-stage novel object handling methodology overview

This research poses grasp synthesis as a search problem—find a location that will facilitate the effective and stable handling of an object from a potentially infinite set. An object-agnostic, sample-based methodology is proposed that consists of three major stages, each driven by a learning component. Figure 4.1 illustrates the conceptual stages of the proposed method.

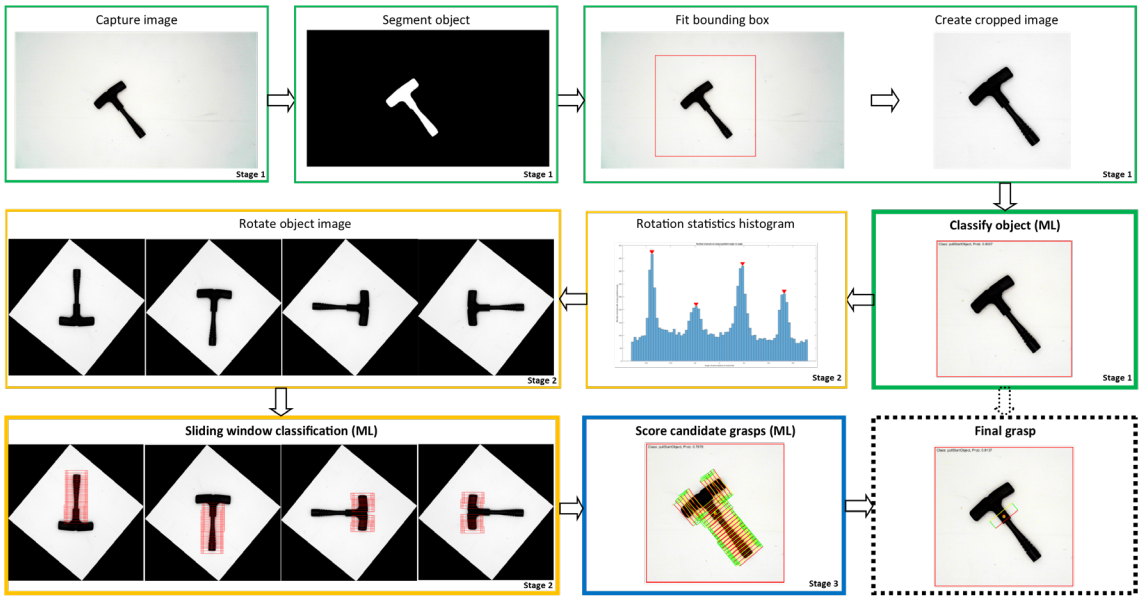


Figure 4.1. **Top-level process diagram of the proposed 3-stage grasping methodology. The first, second and third stages are denoted by green, yellow and blue, respectively. First, an object is captured and segmented from its background. Second, a bounding box is fitted and classified. If classification strength is low, the object is rotated according to major orientations within the image. Rotated images can then be used to generate numerous candidate grasps, of which a final grasp is selected and communicated to the robotic manipulator.**

Stage 1 locates, classifies and catalogues objects present within the workspace through vision. Objects are segmented and prepared for classification using DIP. In the case that an object within the workspace is identified with a high degree of confidence and coupled with a well-established grasping point, this confirmed grasp may be implemented directly, completing the task. In all other cases, further processing is required to compute a grasp appropriate for implementation.

Stage 2 generates the pose and orientation of many hypothesis grasps. Local maxima within an orientational histogram containing statistics on the gradient directions of each pixel, may be used to rotate the image according to major angles within the image. A rectangular grasping window corresponding to the dimensions of the gripper is iterated across each rotated image, classified by the stage 2 learner. The orientation and pose of windows classified as a potential grasping area are transformed into the original image space and

recorded. This is a binary classification network, labelling input samples as either belonging to the *positiveGrasp* class or *negativeGrasp* class.

Stage 3 integrates various sources of information. Though the number of inputs is limited within this dissertation, it is intended that the stage 3 framework is such that varying sensors or known object properties may serve as inputs if they are available, e.g., a secondary view of the object, centre of gravity of the object, scores related to each hypothesis grasping window, temperature, material properties, friction coefficients, gripper feedback, weight, etc. With the proliferation of networked IoT devices, measuring many such properties is becoming increasingly accessible. The stage 3 learner uses such sources of information to score each candidate grasp within the hypothesis pool. The top-scoring candidate may then be converted to robot space to complete the task. Depending on the outcome, the implemented window, pose, orientation, object class and accompanying properties may be catalogued.

To investigate the feasibility surrounding the related implementational aspects of the proposed methodology, it was necessary to develop an experimental prototype. The prototype consisted of a conveyor belt, load-cells, enclosure, vision system, beam sensor and robotic manipulator. Figures 4.2 and 4.3 show the system employed by this research to illustrate the 3-stage approach. It is intended that objects are placed haphazardly at the end of the conveyor belt furthest from the manipulator. Detected objects are automatically moved through the vision enclosure to within the workspace of the robot, where they may be picked without explicit instruction. The prototype represents a microcosm of the grasping task, though, the conceptual basis of the proposed grasp synthesis method is not dependent on the exact physical properties of such a system. Detailed information regarding implementation and hardware can be found in Chapter 9.

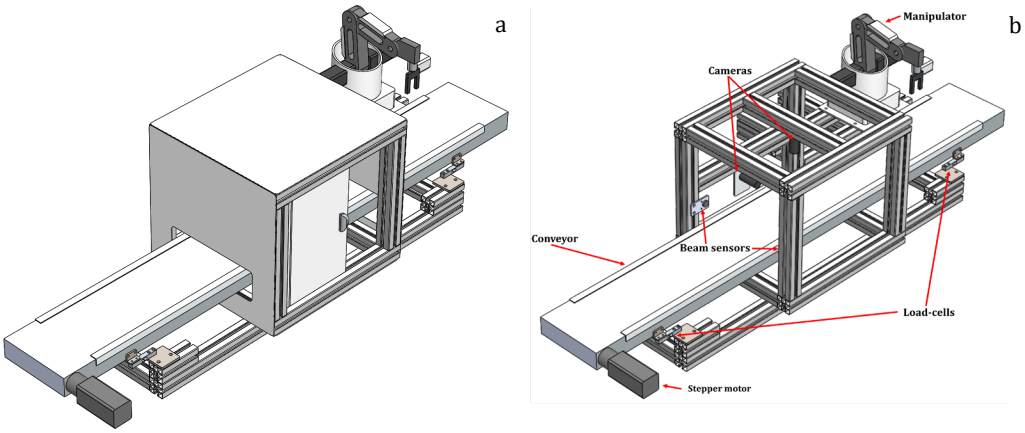


***Figure 4.2*. (a)—isometric view of the proposed prototype. (b)—annotated isometric view of the proposed prototype with covers removed from the vision enclosure.**

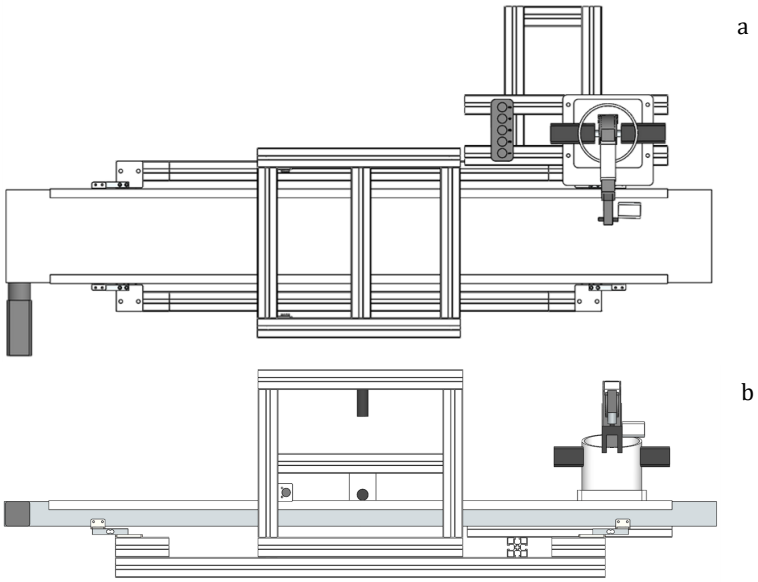


***Figure 4.3*. (a)—top view of the proposed prototype. (b)—front view of the proposed prototype.**

4.2 Representation

Several coordinate frames are considered by the proposed work, graphically illustrated in Figure 4.4. Candidate objects are observed from two separate perspectives through vision, offset by 90° and rotated by 180°. To translate objects across frames, a conveyor belt is employed. Image space and robot space are consolidated by translating imaged objects by a fixed amount between these coordinate frames. By compensating for this set amount, the conveyor frame is effectively eliminated. Thus, image and robot coordinate frames can be treated as though they occupy the same space. The conveyor belt rests on 4 load-cells, which are used for COG measurements. The xy-plane of this coordinate frame is shared by the top camera frame and can be consolidated by a known transform.

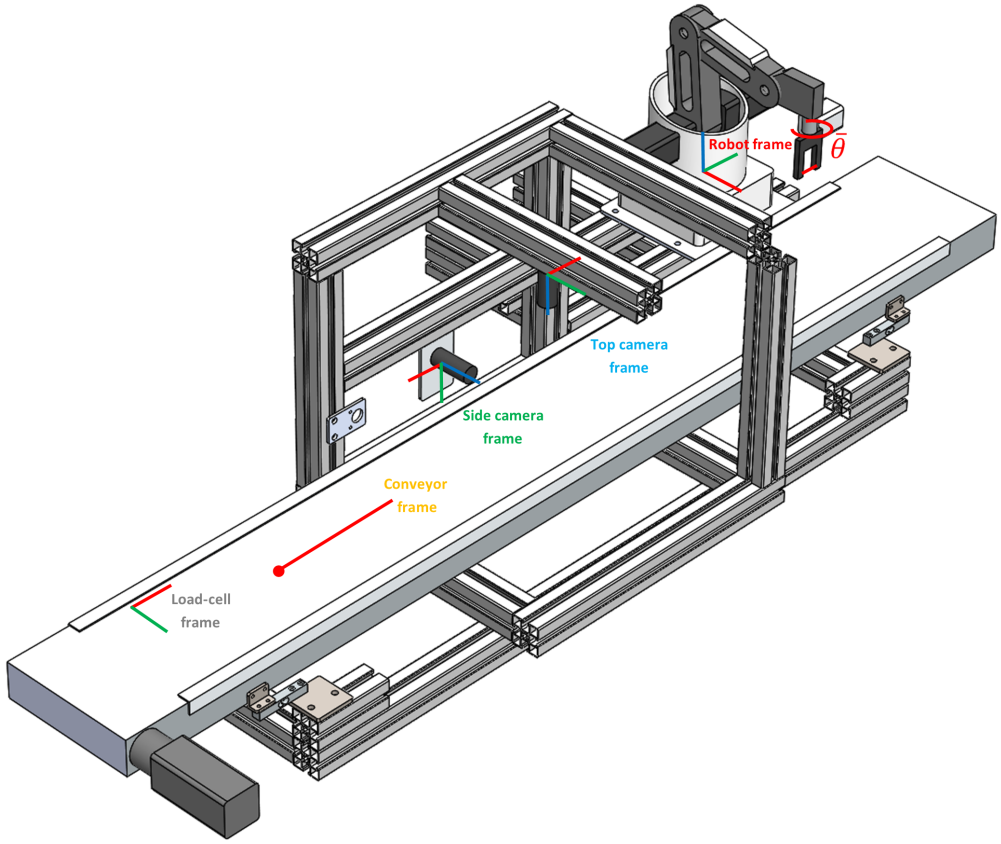


***Figure 4.4*. Visualisation of various coordinate frames associated with the proposed prototype. Red, green and blue express the x-, y- and z-axes, respectively.**

In this thesis, object-agnostic grasp pose generation is considered from two monocular, RGB observations of the scene. Figure 4.5 graphically depicts the associated vision space. All other coordinate frames are considered with respect to this space.

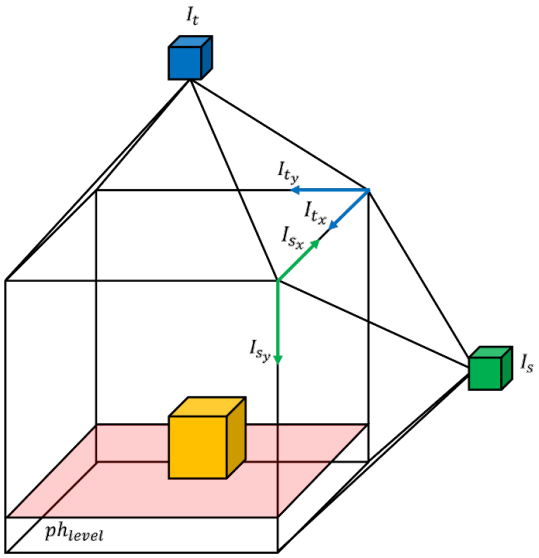


***Figure 4.5*. Illustration of an object imaged from two distinct viewpoints. I_t denotes the 2-dimensional top-view camera space, represented by x-axis I_{t_x} and y-axis I_{t_y}. I_s denotes the 2-dimensional side-view camera space, represented by x-axis I_{s_x} and y-axis I_{s_y}. ph_{level} refers to the height of the conveyor, upon which the object rests.**

Top-view camera space I_t is described by:

$$I_t[x, y] \in \mathbb{R}^{3\times H\times W} \tag{4.1}$$

where $I_t[x, y]$ are real, 3-dimensional pixel values within $\mathbb{R}^{3\times H\times W}$, where H and W are the height and width of the image, respectively. The intrinsic parameters of the camera are known *a priori*. Similarly, the side-view camera space I_s is described by:

$$I_s[x, y] \in \mathbb{R}^{3\times H\times W} \tag{4.2}$$

where $I_s[x, y]$ are real, 3-dimensional pixel values within $\mathbb{R}^{3\times H\times W}$. Two identical sensors are used to generate I_t and I_s, thus H and W are consistent between coordinate frames. Similar to many other works throughout literature [13, 58, 60, 65, 66, 70, 72], antipodal grasps are considered from a perspective perpendicular to a planar surface, given an image of the scene. In this work however, grasp representation has an additional component, associated with the side-view perspective. Within I_t, grasp pose is characterised by a 5-dimensional, rectangular representation—graphically depicted in Figure 4.6—a. Similarly, a 4-dimensional planar representation is assumed within I_s. The side-view rectangle is depicted in Figure 4.6—b.

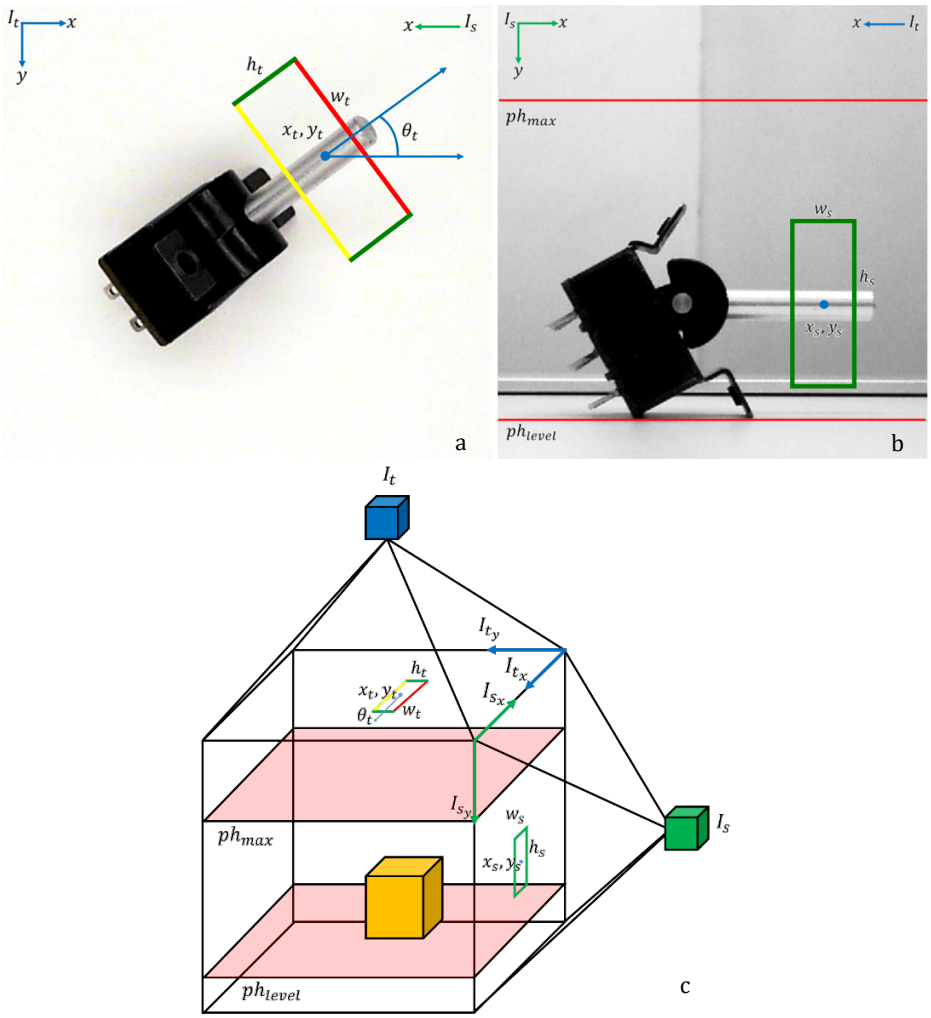


Figure 4.6. **(a)—5-dimenstional top-view grasping rectangle representation. h_t and w_t represent the height and width of the grasping rectangle, respectively. θ_t is the angle relative to the horizontal image x-axis I_{t_x}. x_t, y_t denote the centre coordinates of the rectangle. (b)—4-dimenstional side-view grasping rectangle representation. h_s and w_s represent the height and width of the grasping rectangle, respectively. x_s, y_s are the centre coordinates of the rectangle from the side perspective. ph_{max} and ph_{level} denote the maximum object height supported by the enclosure and the height of the conveyor, respectively. (c)—diagram illustrating the conceptual basis and relative positions of the top-view grasping rectangle representation and the side-view grasping rectangle representation.**

The top-view pose component of a grasp, $pose_t$, is defined as:

$$pose_t = \{x_t, y_t, \theta_t, h_t, w_t\} \in I_t \tag{4.3}$$

where x_t, y_t refer to the gripper's centre position in Cartesian coordinates within I_t. Rotation of the rectangle is defined by angle θ_t about centre point x_t, y_t, with respect to the horizontal axis I_{t_x}. Physically, the 2 fingers of the gripper close perpendicular to θ_t. h_t relates to the physical width of the gripper in pixels and w_t refers to the available distance between the two parallel plates of the gripper at maximum extension, also measured in pixels. Note that h_t and w_t are fixed, as the dimensions of the gripper from perspective I_t do not change. This rectangular grasping window as an image is referred to as I_{RGW}. I_{RGW} is iterated across I_t and assessed for potential graspability by the stage 2 classifier. I_{RGW} directly relates to the physical placement of the gripper in an open position, from a top-down perspective.

The side-view pose component of a grasp, $pose_s$, is defined as:

$$pose_s = \{x_s, y_s, h_s, w_s\} \in I_s \quad \textbf{(4.4)}$$

where x_s, y_s refer to the centre position of the ideal contact area of the gripper pads, within I_s. h_s and w_s are the height and width of this contact area in pixels, respectively, and vary depending on the distance between the object and side-view camera, i.e. h_s and w_s increase in value as I_{t_y} decreases. For more information regarding perspective distortion, refer to the paper by Peng *et al.*, in which perspective distortion is corrected for images of faces [379]. I_{SVW} defines the side-view rectangular window as its own image. I_{SVW} is constrained within the upper limit of object height supported by the enclosure ph_{max} and conveyor level ph_{level}. I_{SVW} physically relates to the maximal contact area of the gripper pads when closed onto an object, from a side-on perspective.

Essentially, $pose_t$ is used to find the $\bar{x}$, $\bar{y}$ and $\bar{\theta}$ components of a grasp in robot space, and $pose_s$ is used to find the $\bar{z}$ component. Combining top-view pose component, $pose_t$, and side-view pose component, $pose_s$, yields multi-view location array g, referred to as a candidate grasping window:

$$g = \{pose_t, pose_s, S_c\} \quad \textbf{(4.5)}$$

where S_c is a matrix containing scores related to the specific grasp candidate g. The scores used as inputs to the stage 3 framework of the proposed methodology are contained in S_c. This matrix is not fixed and may vary depending on the inputs and information available to the system. In this thesis, 10 scores are associated with each grasp:

$$S_c = \{K_{I_{RGW}}, S_{ss}, S_{cs}, S_{lsl}, S_{lsr}, S_{ptp}, S_{COG}, S_{svol}, S_{shs}, S_{sws}\} \quad \textbf{(4.6)}$$

where $K_{I_{RGW}}$ is taken as the softmax function output of the stage 2 CNN for the *positiveGrasp* class, with input I_{RGW}. $K_{I_{RGW}}$ ranges from $[0, 1]$, scoring 1 if the network considers the area perfectly graspable and 0 if the network considers the area not graspable. Figure 4.7 illustrates some examples of the response of this score to various grasping windows, denoted as I_{RCW} during the initial classification stage.

***Figure 4.7*. Examples of I_{RCW} assessed by the stage 2 classifier and the resulting classification strength $K_{I_{RCW}}$. (a)—$K_{I_{RCW}} = 0.5281$. (b)—$K_{I_{RCW}} = 0.0000$. (c)—$K_{I_{RCW}} = 0.991$. (d)—$K_{I_{RCW}} = 0.9995$. (e)—$K_{I_{RCW}} = 0.0004$.**

S_{ss} relates the symmetry of I_{RGW}. S_{ss} is calculated by first mirroring I_{RGW} as per relationship:

$$\begin{bmatrix} x^* \\ y^* \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad \textbf{(4.7)}$$

where x, y are the source coordinates of a given image and x^*, y^* are the destination coordinates. This yields a mirrored version of I_{RGW} denoted as $I_{RGW_{mirror}}$. An absolute difference image $S_{ssI_{diff}}$ is calculated by comparing pixel-wise intensity levels:

$$S_{ssI_{diff}}[x,y] = \left|I_{RGW}[x,y] - I_{RGW_{mirror}}[x,y]\right| \quad \textbf{(4.8)}$$

A binary image of $S_{ssI_{diff}}$ is created by thresholding each pixel:

$$S_{ssI_{Th}}[x,y] = \begin{cases} 1, & if\ S_{ssI_{diff}}[x,y] \geq T_{S_{ssI}} \\ 0, & if\ S_{ssI_{diff}}[x,y] < T_{S_{ssI}} \end{cases} \quad \textbf{(4.9)}$$

where $T_{S_{ssI}}$ is the fixed threshold applied to difference image $S_{ssI_{diff}}$. Finally, S_{ss} quantifies the symmetry of I_{RGW} as a function of the binary difference between I_{RGW} and mirrored version $I_{RGW_{mirror}}$:

$$S_{ss} = 1 - \frac{S_{ssI_{true}}}{S_{ssI_{total}}} \quad \textbf{(4.10)}$$

where $S_{ssI_{true}}$ is the number of pixels within $S_{ssI_{Th}}$ that register a value of 1 and $S_{ssI_{total}}$ is the total number of pixels within $S_{ssI_{Th}}$:

$$S_{ssI_{true}} = \sum_{x=1}^{W_{I_{RGW}}} \sum_{y=1}^{H_{I_{RGW}}} S_{ssI_{Th}}[x,y], \quad if\ S_{ssI_{Th}}[x,y] = 1 \quad \textbf{(4.11)}$$

$$S_{ssI_{total}} = \sum_{x=1}^{W_{I_{RGW}}} \sum_{y=1}^{H_{I_{RGW}}} S_{ssI_{Th}}[x,y] \quad \textbf{(4.12)}$$

where $H_{I_{RGW}}$ and $W_{I_{RGW}}$ are the height and width of I_{RGW} in pixels, respectively. S_{ss} ranges from $[0, 1]$, approaching 1 as the image is perfectly symmetrical. Graphical representations of the various steps used to calculate S_{ss} are illustrated in Figure 4.8.

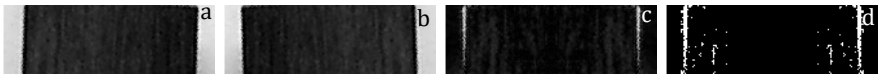


Figure 4.8. **(a)—assessed grasping rectangle I_{RGW}. (b)—mirrored grasping rectangle $I_{RGW_{mirror}}$. (c)—absolute difference image $S_{ssI_{diff}}$. (d)—binary image $S_{ssI_{Th}}$.**

S_{cs} scores how closely the centre of the graspable area within I_{RGW} is situated to the centre of I_{RGW}. S_{lsl} and S_{lsr} score the strength of the left and right edges of the grasp area, respectively. S_{cs}, S_{lsl} and S_{lsr} are calculated using binary image $I_{RGW_{edge}}$:

$$I_{RGW_{edge}}[x,y] = \begin{cases} 1, & if\ S_{csI_{sobel}}[x,y] \geq T_{S_{cs}} \\ 0, & if\ S_{csI_{sobel}}[x,y] < T_{S_{cs}} \end{cases} \quad \textbf{(4.13)}$$

where $T_{S_{cs}}$ is the threshold applied to gradient magnitude image $S_{csI_{sobel}}$. $S_{csI_{sobel}}$ is created by filtering I_{RGW} with a standard Sobel filter in the y-direction only:

$$gradient_y[x,y] = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * A \quad \textbf{(4.14)}$$

where A is the input image. More information regarding Sobel filters and spatial filtering in general, is presented in Gonzalez and Woods [247]. An array counting the number of pixels within the y-axis that register a value of 1, along the x-axis of $I_{RGW_{edge}}$, is tabulated:

$$S_{csI_{sobelpeaks}}[x] = \sum_{y=1}^{H_{I_{RGW}}} I_{RGW_{edge}}[y] \quad if\ I_{RGW_{edge}}[y] = 1 \tag{4.15}$$

$S_{csI_{sobelpeaks}}$ may then be used to find the peaks corresponding to strong vertical edges within $I_{RGW_{edge}}$. The centre location, left-most peak location and right-most peak location along the x-axis are denoted as $S_{cs_{pk}}$, $S_{lsl_{pk}}$ and $S_{lsr_{pk}}$, respectively. The number of pixels within peaks $S_{lsl_{pk}}$ and $S_{lsr_{pk}}$ are denoted as $S_{lsl_{pkN}}$ and $S_{lsr_{pkN}}$, respectively. A graphical representation of this array is illustrated in Figure 4.9.

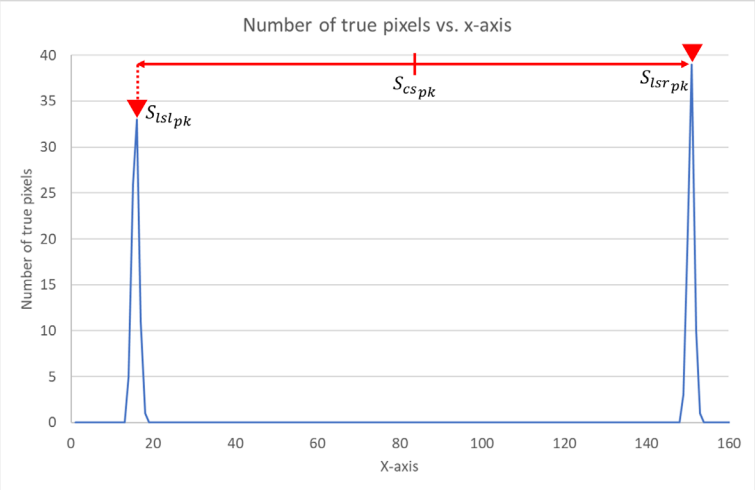


***Figure 4.9*. Illustration of $S_{csI_{sobelpeaks}}$ array. $S_{cs_{pk}}$, $S_{lsl_{pk}}$ *and* $S_{lsr_{pk}}$ have been annotated. This graph represents the total number of pixels that register a value of 1 within the y-axis, along the x-axis.**

From these values, S_{cs}, S_{lsl} and S_{lsr} can be calculated:

$$S_{cs} = 1 - \frac{\left| \frac{W_{I_{RGW}}}{2} - \frac{S_{lsl_{pk}} + S_{lsr_{pk}}}{2} \right|}{\frac{W_{I_{RGW}}}{2}} \tag{4.16}$$

$$S_{lsl} = \frac{S_{lsl_{pkN}}}{H_{I_{RGW}}} \tag{4.17}$$

$$S_{lsr} = \frac{S_{lsr_{pkN}}}{H_{I_{RGW}}} \tag{4.18}$$

S_{cs}, S_{lsl} and S_{lsr} range in value from $[0, 1]$. S_{cs} scores 1 if the middle value between $S_{lsl_{pk}}$ and $S_{lsr_{pk}}$ is perfectly situated at the centre of I_{RGW}. S_{lsl} and S_{lsr} approach values of 1 as the edges within $I_{RGW_{edge}}$ becomes less noisy and perfectly vertical, e.g. the jagged edges of a bolt may score low, while the straight edge of a pencil may score high. Figure 4.10 illustrates

several examples. S_{lsl} and S_{lsr} result in 1 if the vertical lines within $I_{RGW_{edge}}$ are completely unbroken from top to bottom.

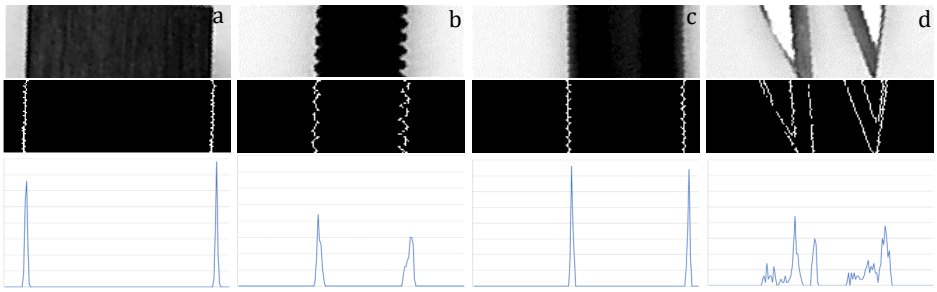


***Figure 4.10*. Graphical illustration of input I_{RGW} (top), output $I_{RGW_{edge}}$ (middle) and resulting pixel array $S_{csI_{sobelpeaks}}$ (bottom). Objects with straight and well-defined edges will generate high $S_{lsl_{pk}}$ *and* $S_{lsr_{pk}}$ scores. (a)—grasping area that is centre with strong edges. (b)—grasping area that is centre with weak edges. (c)—grasping area with strong edges that is off-centre. (d)—grasping area classified as a *negativeGrasp* class with inappropriate geometry.**

S_{ptp} is a simple score that relays the proportion of I_{RGW} occupied by the object:

$$S_{ptp} = \frac{S_{ptp_{true}}}{S_{ptp_{total}}} \tag{4.19}$$

where $S_{ptp_{true}}$ is the number of pixels within thresholded binary image $I_{RGW_{Th}}$ that register a value of 1 and $S_{ptp_{total}}$ is the total number of pixels within $I_{RGW_{Th}}$:

$$S_{ptp_{true}} = \sum_{x=1}^{W_{I_{RGW}}} \sum_{y=1}^{H_{I_{RGW}}} I_{RGW_{Th}}[x,y], \quad if\ I_{RGW_{Th}}[x,y] = 1 \tag{4.20}$$

$$S_{ptp_{total}} = \sum_{x=1}^{W_{I_{RGW}}} \sum_{y=1}^{H_{I_{RGW}}} I_{RGW_{Th}}[x,y] \tag{4.21}$$

where $I_{RGW_{Th}}$ is created by thresholding I_{RGW}:

$$I_{RGW_{Th}}[x,y] = \begin{cases} 1, & if\ I_{RGW}[x,y] \geq T_t \\ 0, & if\ I_{RGW}[x,y] < T_t \end{cases} \tag{4.22}$$

where T_t is the fixed threshold applied to I_t when segmenting an object from the workspace background. More information on this process is available in Section 4.3. S_{ptp} ranges from $[0, 1]$, resulting in a score of 1 if the entire grasping window I_{RGW} is occupied by the object, and 0 if there is no detected object present within I_{RGW}. Figure 4.11 illustrates the response of this score to various grasping windows.

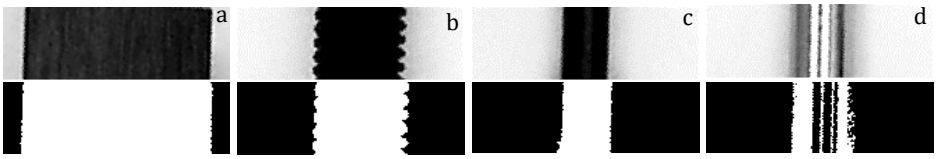


***Figure 4.11*. Graphical illustration of input I_{RGW} (top) and output $I_{RGW_{Th}}$ (bottom). (a)—S_{ptp} = 0.8304. (b)—S_{ptp} = 0.3995. (c)—S_{ptp} = 0.2168. (d)—S_{ptp} = 0.1653.**

S_{COG} scores the distance from I_{RGW} centre x_t, y_t to the COG position of an object. The coordinates of the COG of an object within top-view image space I_t are denoted as $x_{t_{COG}}, y_{t_{COG}}$. To calculate these coordinates, 4 load-cell measurements are used. Figure 4.12 illustrates the location of the load-cells within the prototype and the corresponding load-cell coordinate frame.

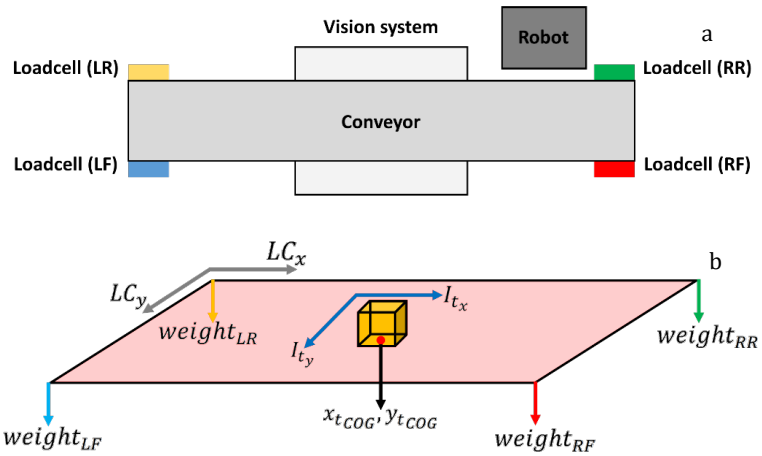


Figure 4.12. **(a)—load-cell configuration. The employed conveyor belt rests on 4 load-cells to measure the COG of single objects presented to the system. (b)—graphical illustration of the load-cell arrangement and related coordinate frames.**

To facilitate the consolidation between load-cell space LC and top-view image space I_t, two transformations are computed:

$$x_{t_{COG}} = tr_{LCX}(coeffx) \tag{4.23}$$

$$y_{t_{COG}} = tr_{LCY}(coeffy) \tag{4.24}$$

where tr_{LCX} and tr_{LCY} are transforms found through experimentation that relate coefficients calculated from load-cell measurements and $I_t[x, y]$. For more information regarding physical implementation details, see Section 9.3.4. $coeffx$ and $coeffy$ are calculated as ratios of weight distribution measured by the 4 load-cells:

$$coeffx = \frac{weight_{RF} + weight_{RR}}{weight_{total}} \tag{4.25}$$

$$coeffy = \frac{weight_{RF} + weight_{LF}}{weight_{total}} \tag{4.26}$$

where $weight_{total}$ is calculated as the sum of load-cell measurements:

$$weight_{total} = weight_{LR} + weight_{RR} + weight_{LF} + weight_{RF} \tag{4.27}$$

where $weight_{LR}, weight_{RR}, weight_{LF}$ and $weight_{RF}$ are real load-cell measurements corresponding to the locations depicted in Figure 4.12. The coordinates of the COG of an object within the load-cell coordinate frame $LC[x, y]$ can be calculated, but is avoided in this work. For a more in-depth description of this method, refer to Patel and Topiwala [380].

S_{COG} is calculated as the proportion of allowable distance between I_{RGW} centre x_t, y_t and COG position $x_{t_{COG}}, y_{t_{COG}}$:

$$S_{COG} = 1 - \frac{\sqrt{\left(x_{t_{COG}} - x_t\right)^2 + \left(y_{t_{COG}} - y_t\right)^2}}{S_{COG_{distMAX}}} \quad \textbf{(4.28)}$$

where $S_{COG_{distMAX}}$ represents the maximal allowable distance between x_t, y_t and $x_{t_{COG}}, y_{t_{COG}}$. $S_{COG_{distMAX}}$ is fixed according to window size. S_{COG} ranges in value from $[0, 1]$, scoring 1 if the object COG and I_{RGW} centre share the same coordinates. If the calculated distance exceeds $S_{COG_{distMAX}}$, S_{COG} defaults to 0. It should be noted that $x_{t_{COG}}, y_{t_{COG}}$ may be approximated through vision, thus avoiding load-cell measurements altogether. This may be achieved by calculating the centre of mass of a binary image of the object within I_t—refer to x_o, y_o calculations in Section 4.3 for more information. Figure 4.13 graphically depicts the distance used to score S_{COG}.

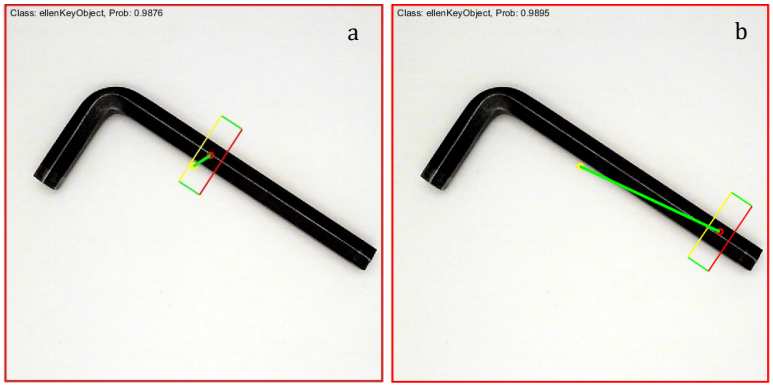


Figure 4.13. **Illustration of COG distance. The yellow circle represents $x_{t_{COG}}, y_{t_{COG}}$, found through vision. The red circle represents I_{RGW} *centre* x_t, y_t. (a)—example of high scoring S_{COG}. (b)—example of low scoring S_{COG}.**

The final three scores, S_{svol}, S_{shs} and S_{sws}, are calculated using I_{SVW} within side-view image space I_s. The side-view overlap score S_{svol} is calculated as the proportion of I_{SVW} occupied by the object. This score calculation is identical to S_{ptp}, but utilises I_{SVW} instead of I_{RGW}. Similarly, the vertical-symmetry score S_{shs} calculation is identical to the procedure used in S_{cs}. The horizontal-symmetry score S_{sws} calculation is also identical to S_{cs}, however, the x- and y-axes are flipped. S_{svol}, S_{shs} and S_{sws} range from [0, 1]. Figure 4.14 shows an example of I_{SVW} extraction from I_s. For more information regarding I_{SVW} generation, see Section 4.3.

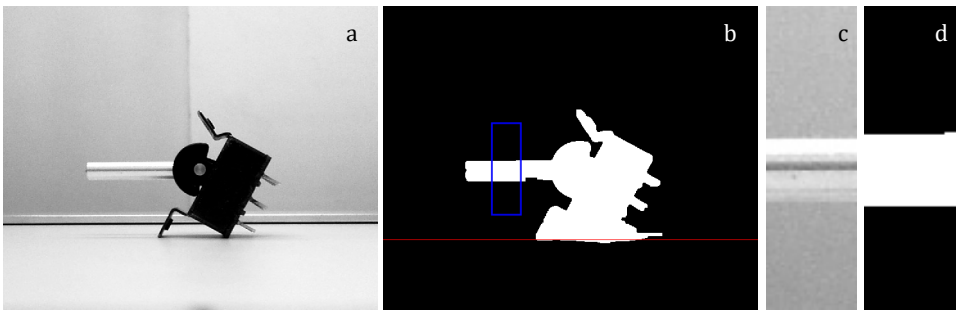


Figure 4.14. **Example of I_{SVW} extraction and threshold from I_s. (a)—I_s. (b)—binary version of I_s, where ph_{level} (red) and I_{SVW} (blue) are annotated. (c)—I_{SVW}. (d)—binary version of I_{SVW}.**

4.3 Grasp pose generation and selection

In this dissertation, grasp synthesis is undertaken using a generate-and-test structure, paired with supervised learning. A qualitative process description of the proposed methodology is presented in Chapter 4.1. Details regarding coordinate frames, grasp representation and the score matrix are noted in Chapter 4.2.

To generate a pool of hypothesis grasps, an image of the scene containing the candidate object is captured in top-view camera space I_t. Simultaneously, side-view camera space I_s captures a second perspective of the scene. Prior to processing, I_t and I_s are converted to greyscale using the Rec. 601 conversion, resulting in single-dimension, 8-bit pixel values that range from $[0\,..\,255]$. For more information related to this standard, the reader is directed to the original protocol proposed by D. Tynan in 1998 [381]. RGB data is ignored to reduce the input size for the employed CNNs—improving performance. For more information regarding the correlation between the number of input channels and computational efficiency, refer to Goodfellow, Bengio and Courville [135] or Raschka and Mirjalili [206]. Moreover, in this work, it was observed that greyscale conversion significantly improved the computational performance of the employed CNNs without significantly affecting classification accuracy. Objects within I_t are segmented via binary threshold:

$$I_{t_{Th}}[x,y] = \begin{cases} 1, & if\ I_t[x,y] \geq T_t \\ 0, & if\ I_t[x,y] < T_t \end{cases} \tag{4.29}$$

where x and y range from $[1\,..\,W]$ and $[1\,..\,H]$, respectively. T_t is the fixed threshold level applied to I_t to create binary image $I_{t_{Th}}$. T_t was found through experimentation and could be effectively implemented as a fixed value because of the enclosure design of the prototype—detailed in Section 9.3. Several minor morphological filters are applied to $I_{t_{Th}}$ for noise reduction. Gonzalez and Woods present an excellent resource on this topic [247]. The centroid of the candidate object x_o, y_o within $I_{t_{Th}}$ is calculated via centre of mass calculation:

$$x_o = \frac{\sum_{x=1}^{W}\sum_{y=1}^{H} xI_{t_{Th}}[x,y]}{N_{pix}} \tag{4.30}$$

$$y_o = \frac{\sum_{x=1}^{W}\sum_{y=1}^{H} yI_{t_{Th}}[x,y]}{N_{pix}} \tag{4.31}$$

where N_{pix} is the total number of pixels within $I_{t_{Th}}$ with a value of 1:

$$N_{pix} = \sum_{x=1}^{W}\sum_{y=1}^{H} I_{t_{Th}}[x,y]\,, \quad if\ I_{t_{Th}}[x,y] = 1 \tag{4.32}$$

N_{pix} also represents the area of the segmented object, in pixels. It should be noted that I_t and $I_{t_{Th}}$ share the same coordinate space. A fixed-size, square bounding box is fitted with centre coordinates x_o, y_o of size $H_{I_{BB}} \times H_{I_{BB}}$ within the top-view coordinate frame I_t. $H_{I_{BB}}$ is constrained by the desired input size of the stage 1 learner and consequently, limits the allowable object size. This bounding box as an image is referred to as I_{BB}. For a graphical illustration of the process by which I_{BB} is found, refer to Figure 4.15.

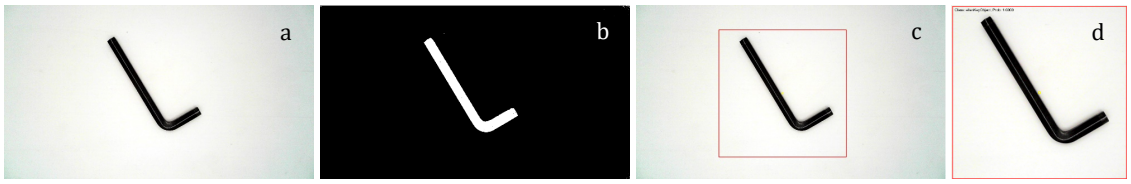


***Figure 4.15*. (a)—input top-view image I_t. (b)—thresholded image I_{tTh}. (c)—object bounding box I_{BB} overlaid onto I_t. (d)—classified I_{BB}.**

A downscaled version of I_{BB} is classified by the stage 1 CNN. The resulting softmax function output is denoted as $K_{I_{BB}}$. This step determines whether the object within I_t is considered known or unknown. If the $K_{I_{BB}}$ value is equal to or above the classification acceptance threshold T_{class}, the object is considered known and therefore, a pre-determined grasp may be fitted directly. If $K_{I_{BB}}$ is less than T_{class}, a matrix containing a set of top-view grasp pose hypotheses is populated:

$$pose_{t_{1\,..\,m}} = \begin{Bmatrix} x_{t_1}, y_{t_1}, \theta_{t_1}, h_t, w_t \\ \vdots \\ x_{t_n}, y_{t_n}, \theta_{t_n}, h_t, w_t \\ \vdots \\ x_{t_m}, y_{t_m}, \theta_{t_m}, h_t, w_t \end{Bmatrix} \tag{4.33}$$

where n ranges from $[1\,..\,m]$. h_t and w_t are fixed according to the maximal width of the gripper when opened and relate to top-view rectangular grasping window I_{RGW}. The angle of a candidate grasp θ_{t_n} within pose matrix $pose_{t_{1\,..\,m}}$ is found by rotating I_{BB} according to the local maxima within an orientational histogram, which is collated as the number of pixels counted at each unique gradient angle resulting from Sobel filter:

$$gradient_x[x,y] = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * A \tag{4.34}$$

$$gradient_y[x,y] = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * A \tag{4.35}$$

$$\theta_{gradient}[x,y] = arctan\left(\frac{gradient_y}{gradient_x}\right) \tag{4.36}$$

where $gradient_x$ and $gradient_y$ are the horizontal and vertical gradient values of a pixel, respectively. $\theta_{gradient}$ is the resultant gradient direction. This filter responds to strong edges and major orientations within an image. Figure 4.16 graphically illustrates a typical histogram output, given input I_{BB}. The gradient angle filter $\theta_{gradient}[x,y]$ is applied to I_{BB}. The resulting 2-dimenstional matrix contains the gradient directions of each pixel within I_{BB}, defined as θ_{Gdir}:

$$\theta_{Gdir}[x,y] = \theta_{gradient}(I_{BB}[x,y]) \tag{4.37}$$

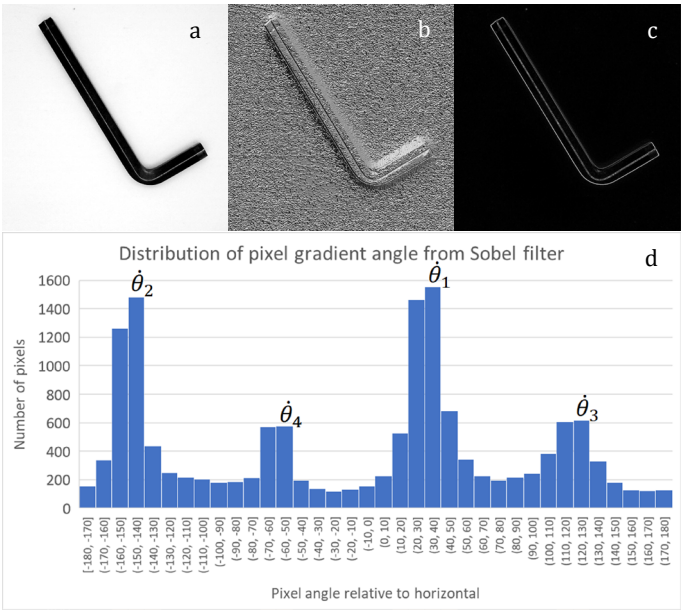


Figure 4.16. **(a)—input top-view, bounding box image I_{BB}. (b)—resulting pixel gradients from Sobel filter θ_{Gdir}. (c)—resulting pixel magnitudes from Sobel filter. (d)—resulting orientational histogram, where local maxima represent strong orientations within the filtered image.**

Statistics related to θ_{Gdir} are tabulated in the form of a histogram that counts the number of occurrences of pixel angle $\theta_{gradient}$ within bins of interval n_{bin} of range $[-180° \,..\, 180°]$. The top 4 maxima from θ_{Gdir} are noted within rotation array θ_{Sobel}:

$$\theta_{Sobel} = \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \end{pmatrix} \tag{4.38}$$

where $\dot{\theta}_1$ denotes the highest number of pixels counted at a specific angle and $\dot{\theta}_4$ denotes the 4th highest. I_{BB} is rotated about centre coordinates x_o, y_o for each angle within θ_{Sobel}, $\dot{\theta}_{1\,..\,4}$, resulting in 4 rotated versions of I_{BB} denoted as $I_{BBR_{1\,..\,4}}$. Figure 4.17 illustrates an example of this process. Cartesian coordinates within rotated space I_{BBR} are denoted as $I_{BBR}[i, j]$. It should be noted that the height and width of $I_{BBR_{1\,..\,4}}$, $H_{I_{BBR}}$ and $W_{I_{BBR}}$, respectively, are dependent on $\dot{\theta}_{1\,..\,4}$, i.e. the degree to which the image has been rotated. Sun, Yu, Liu and Gu employ a similar technique to detect strong orientations of objects within an image [66]. They selected the top 5 orientations within their histogram. In this work, I_{BB} is only rotated by the top 4 maxima within θ_{Gdir} to increase efficiency by reducing the effective search space of the proposed methodology. Moreover, this value may be varied depending on the desired search resolution.

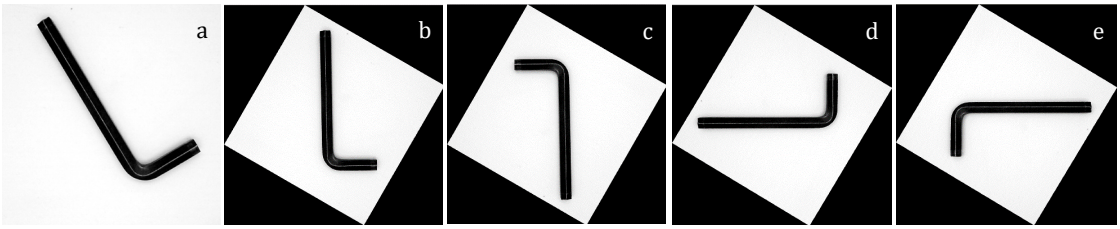


Figure 4.17. **(a)— input top-view, bounding box image I_{BB}. (b), (c), (d), (e)—examples of I_{BB} rotated by $\dot{\theta}_{1\,..\,4}$.**

A rectangular classification window I_{RCW_n} of height h_t and width w_t and centre i_{RCW}, j_{RCW} is iterated across each rotated bounding box image $I_{BBR_{1\,..\,4}}[i,j]$ within horizontal interval $\left[i_1 + \frac{w_t}{2}, \ldots, i_{W_{I_{BBR}}} - \frac{w_t}{2}\right]$ and vertical interval $\left[j_1 + \frac{h_t}{2}, \ldots, j_{H_{I_{BBR}}} - \frac{h_t}{2}\right]$. Horizontal and vertical step magnitudes are denoted as i_{step} and j_{step}, respectively. Note that $I_{t_{Th}}$ may be used to reduce this search area by only considering I_{RCW_n} in which the object is present. I_{RCW_n} represents an area that may be appropriate for grasping. At each stepped location $I_{BBR_{1\,..\,4}}[i_{RCW}, j_{RCW}]$, I_{RCW_n} is assessed by the stage 2 classifier. The softmax function output for each assessed rectangle is denoted as $K_{I_{RCW_n}}$. If $K_{I_{RCW_n}}$ is higher than the acceptance threshold T_{grasp} and the sample is labelled as positive, then I_{RCW_n} is considered a hypothesis or candidate grasp. The associated centre location i_{RCW}, j_{RCW} of I_{RCW_n} within rotated space $I_{BBR_{1\,..\,4}}[i,j]$ is transformed to the respective, non-rotated space $I_{BB}[x,y]$, which shares a coordinate space with $I_t[x,y]$. A classified hypothesis grasp within non-rotated space I_t is denoted as a rectangular grasping window I_{RGW_n}. Thus, the centre location of a candidate grasp x_{t_n}, y_{t_n} within pose matrix $pose_t$ is known. For the specific candidate, θ_{t_n} is taken as the respective angle $\dot{\theta}_{1\,..\,4}$ at classification time. The softmax output for each candidate grasp is recorded as $K_{I_{RGW_n}}$. Note that $K_{I_{RGW_n}}$ is added to score matrix S_{c_n}. Additionally, the remaining top-view scores $S_{ss_n}, S_{cs_n}, S_{lsl_n}, S_{lsr_n}, S_{ptp_n}, S_{COG_n}$ are also associated with their respective candidate grasp I_{RGW_n}. Figure 4.18 illustrates examples of classified windows I_{RCW_n} within rotated space $I_{BBR_{1\,..\,4}}[i,j]$ and Figure 4.19 illustrates the resultant hypothesis grasps I_{RGW_n} within I_t.

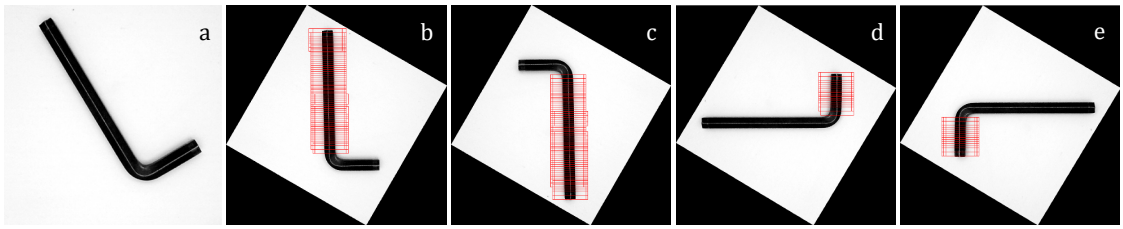


Figure 4.18. **(a)— input top-view, bounding box image I_{BB}. (b, c, d, e)—examples of I_{BB} rotated by $\dot{\theta}_{1\,..\,4}$. The annotated red rectangles represent classified candidate grasping windows I_{RCW_n} with $K_{I_{RCW_n}}$ values higher than T_{grasp}.**

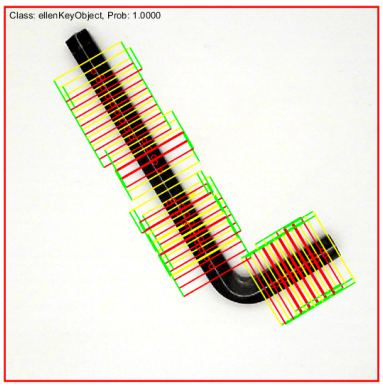


Figure 4.19. **Illustration of classified candidate grasping windows I_{RGW_n} within I_{BB}.**

At this stage, the top-view grasp hypothesis pose matrix $pose_t$ is populated. The side-view camera space I_s may now be addressed. Segmentation from this perspective utilises an image taken of the scene prior to object introduction. This image shares a coordinate frame

with I_s and is denoted as $I_{s_{blank}}$. To improve segmentation robustness, slight noise is added to I_s and $I_{s_{blank}}$ in the form of Gaussian blur:

$$GB[x,y] = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{4.39}$$

where σ is the standard deviation of the Gaussian distribution. The blurred versions of I_s and $I_{s_{blank}}$ are denoted as $I_{s_{GB}}$ and $I_{s_{blank_{GB}}}$, respectively. For a quantitative and qualitative description of the effects of a 2-dimensional Gaussian function on images, refer to Gedraite and Hadad [382]. An absolute difference image $I_{s_{diff}}$ is calculated by comparing the pixel-wise intensity levels between $I_{s_{GB}}$ and $I_{s_{blank_{GB}}}$:

$$I_{s_{diff}}[x,y] = \left| I_{s_{GB}}[x,y] - I_{s_{blank_{GB}}}[x,y] \right| \tag{4.40}$$

The addition of noise prior to image differencing is a common technique used to enhance the quality of background subtraction methodologies [383-385]. A binary image $I_{s_{Th}}$ is created by thresholding each pixel within $I_{s_{diff}}$:

$$I_{s_{Th}}[x,y] = \begin{cases} 1, & if\ I_{s_{diff}}[x,y] \geq T_{I_s} \\ 0, & if\ I_{s_{diff}}[x,y] < T_{I_s} \end{cases} \tag{4.41}$$

where T_{I_s} is a fixed threshold applied to difference image $I_{s_{diff}}$. Several morphological filters are applied to $I_{s_{Th}}$ for noise reduction and improved edge definition. Figure 4.20 visually illustrates the process of segmenting an object within I_s using reference image $I_{s_{blank}}$.

Figure 4.20. **Example of the side-view segmentation process. (a)—reference image $I_{s_{blank}}$ captured prior to object introduction. (b)—side-view image I_s captured post object introduction. (c)—thresholded difference image $I_{s_{Th}}$.**

The segmented image $I_{s_{Th}}$ is used to populate a matrix containing the side-view pose components of grasps $pose_s$ associated with the top-view candidate set $pose_t$. $pose_s$ is defined as:

$$pose_{s_{1\,..\,m}} = \begin{Bmatrix} x_{s_1}, y_{s_1}, h_{s_1}, w_{s_1} \\ \vdots \\ x_{s_n}, y_{s_n}, h_{s_n}, w_{s_n} \\ \vdots \\ x_{s_m}, y_{s_m}, h_{s_m}, w_{s_m} \end{Bmatrix} \tag{4.42}$$

where side-view rectangle width w_{s_n} and height h_{s_n} vary based on object-camera distance and relate to side-view gripper contact area I_{SVW}. This distance is measured by the top-view

y-axis I_{t_y}. Chapter 4.2 gives more information regarding coordinate frames. Consequently, w_{s_n} can be approximated by:

$$w_{s_n} = tr_{sideW}(y_{t_n}) \tag{4.43}$$

where tr_{sideW} is a transformation found to consolidate rectangle width w_{s_n} and the top-view y-axis location of candidate grasp y_{t_n} through testing. Refer to Section 9.3.3 for more details. The ratio of the rectangular side-view contact area does not change. Moreover, this ratio is shared by the ratio of the top-view rectangle. As such, h_{s_n} may be calculated as ratio:

$$h_{s_n} = w_{s_n}\left(\frac{h_t}{w_t}\right) \tag{4.44}$$

Since I_{t_x} and I_{s_x} operate in the same plane, the side-view rectangle x-position x_{s_n} can be approximated as a function of the top-view rectangle x_{t_n} position:

$$x_{s_n} = tr_{sideX}(x_{t_n}) \tag{4.45}$$

where transformation tr_{sideX} was found to relate x_{s_n} and x_{t_n}. A side-view rectangular assessment window I_{SRW_n} of height h_{s_n} and width w_{s_n} at centre position $\left[x_{s_n}, y_{s_{step}}\right]$ is iterated vertically within $I_{s_{Th}}$ along y-axis $y_{s_{step}}$ interval $\left[ph_{max} \,..\, ph_{level} - \frac{h_{s_n}}{2}\right]$. It should be noted that $I_{s_{Th}}$ is used to reduce this search space by only assessing windows I_{SRW_n} in which the object is present. ph_{level} is the height of the platform at the point of object-conveyor contact, annotated in Figure 4.14—b, Section 4.2. ph_{level} falls within side-view coordinate frame I_s and can be estimated using the top-view y-axis object centre location, y_o:

$$ph_{level} = tr_{ph}(y_o) \tag{4.46}$$

where tr_{ph} is a transform that relates the platform level ph_{level} within I_s to the top-view object centre position y_o. ph_{max} is the maximum height of an object within I_s supported by the enclosure, in pixels. At each $y_{s_{step}}$ iteration of window I_{SRW_n}, side-view scores, S_{svol_n}, S_{shs_n} and S_{sws_n}, are calculated. For more information regarding score matrix S_{c_n}, see Section 4.2. A tally of the sum of these scores for each I_{SRW_n} iterated at specific x_{s_n} is calculated as:

$$I_{SRW_{n_{SC}}} = S_{svol_n} + S_{shs_n} + S_{sws_n} \tag{4.47}$$

y_{s_n} is taken as the respective $y_{s_{step}}$ value for the window I_{SRW_n} with the greatest sum $I_{SRW_{n_{SC}}}$ associated with top-view candidate grasp pose $pose_{t_n}$, iterated vertically at position x_{s_n} within $I_{s_{Th}}$. This window as an image is referred to as the side-view hypothesis component of a grasp, I_{SVW_n}. Figure 4.21 illustrates the process by which many I_{SVW_n} that correspond to $pose_{t_n}$ are found.

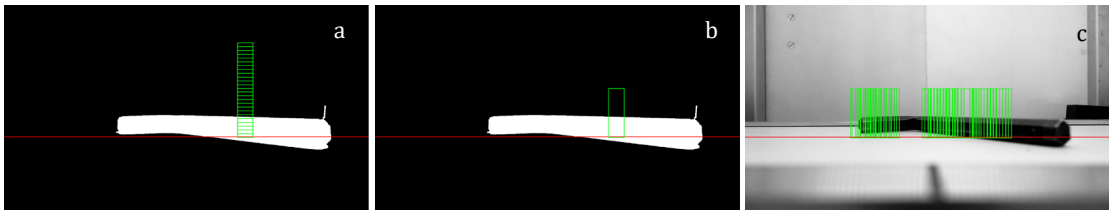


***Figure 4.21*. Illustration of side-view grasp component I_{SVW_n} generation. (a)—$I_{s_{Th}}$ with ph_{level} annotated in red and many assessment windows I_{SRW_n} annotated in green. (b)—$I_{s_{Th}}$ annotated with assessment window I_{SRW_n} at current x_{s_n} with greatest sum $I_{SRW_{n_{SC}}}$. (c)—side-view pose windows I_{SVW_n} overlaid onto I_s.**

Now that the top-view grasp pose hypothesis matrix $pose_{t_{1\,..\,m}}$, the side-view grasp pose hypothesis matrix $pose_{s_{1\,..\,m}}$ and the related score matrix $S_{c_{1\,..\,m}}$ have been populated, these matrices may be collated as the candidate grasp matrix:

$$g_{1\,..\,m} = \{pose_{t_{1\,..\,m}}, pose_{s_{1\,..\,m}}, S_{c_{1\,..\,m}}\} \tag{4.48}$$

$g_{1\,..\,m}$ describes the pool of candidate grasps generated for a specific object in terms of the associated top-view pose component, side-view pose component and candidate scores. This matrix is used by the stage 3 framework to predict two output scores, OS_{pred_n} and OE_{pred_n}, for each candidate grasp g_n from input S_{c_n}. Effectively, the stage 3 component attempts to predict the physical outcome of a candidate grasp g_n in terms of similarity scores OS and OE, based on the score matrix S_{c_n} associated with the considered grasp. Similarity scores are described in Chapter 3.2. Note that other output scores may also be predicted and selected for using this score matrix. The resultant scores associated with each hypothesis grasp g_n are collated into a selection stage grasp set:

$$\dot{g}_{1\,..\,m} = \{pose_{t_{1\,..\,m}}, pose_{s_{1\,..\,m}}, OS_{pred_{1\,..\,m}}, OE_{pred_{1\,..\,m}}\} \tag{4.49}$$

From $\dot{g}$, a final grasp C is selected. The selected candidate C is chosen based on the highest scoring sum of OS_{pred_n} and OE_{pred_n} of individual samples within $\dot{g}$:

$$C = max\left\{\dot{g}_{1\,..\,m}\{OS_{pred_{1\,..\,m}} + OE_{pred_{1\,..\,m}}\}\right\} \tag{4.50}$$

$$G_C = g\{pose_{t_C}, pose_{s_C}, S_{c_C}\} \tag{4.51}$$

G represents a grasp to be implemented, described by grasp pose in terms of coordinate frames I_t and I_s. Finally, the selected candidate G_C is transformed into robot coordinate frame:

$$\bar{G}[\bar{x}, \bar{y}, \bar{z}, \bar{\theta}] = tr_{RC}(G_C[x_{t_C}, y_{t_C}, y_{s_C}, \theta_{t_C}]) \tag{4.52}$$

where tr_{RC} is a known transformation that consolidates a grasp pose $G[x_t, y_t, y_s, \theta_t]$ within vision space to the grasp pose coordinates within robot workspace $\bar{G}[\bar{x}, \bar{y}, \bar{z}, \bar{\theta}]$. $\bar{x}, \bar{y}$ and $\bar{z}$ are related to the x-, y- and z-axes coordinates of a grasp in robot space $\bar{G}$. $\bar{\theta}$ is the end-effector angle. Because the employed 2-fingered gripper is symmetric around $\pm 90°$, $\bar{\theta}$ ranges from $[0°, 90°]$. Figure 4.22 graphically depicts the process described to this point from top-view perspective I_t.

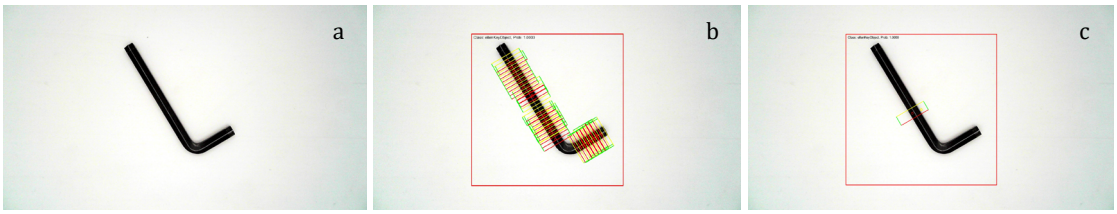


***Figure 4.22*. (a)—top-view image I_t. (b)— classified candidate grasping windows I_{RGW_n} within I_t. (c)— selected grasping window I_{RGW_C} within I_t.**

Chapter 5

Establishing a pool of test objects to evaluate the 3-stage methodology

5.1 Object pool used for training and testing

As mentioned in Chapter 2.6, research within the field of robotic manipulation is currently suffering from a lack of general standardisation, thus, confounding reproducibility. In many cases, authors attempt to mitigate this issue with thorough documentation. Calli *et al.* for instance, record the dimensions, mass, a brief description and images of the objects that make up the YCB object and model set [94]. Works also try to ensure generalised testing of their methodology by collating object test pools that include a wide variety of objects. For this reason, the object pool utilised in this thesis is comprised of 100 objects. Larger objects present in the YCB set, e.g., hammer, soccer ball, pan, spatula, etc., could not be included in the proposed test pool due to the limitations imposed by hardware. This research is focused on industrial small-part assembly. Consequently, the proposed object test set is largely focused on small, intricate objects. The object test pool consists of two groups, objects used to generate training data and objects used for testing.

The proposed object training and test pool is split into 3 series, labelled *known*, *unknown and miscellaneous*. In addition, objects are subcategorised according to utility: common household objects, tools and components. In total, the 100 objects are split into 9 subsets. Note that this pool was curated with respect to minimal redundancy, i.e. selecting objects that sufficiently differ in shape, size, mass-distribution, surface properties, weight, deformability and grasp difficulty. The known series contains 15 unique objects. Training samples for the stage 1, stage 2 and stage 3 networks are generated from the known series exclusively. Figure 5.1 shows the objects within this series and their associated identifier. Table 5.1 is the object register for the known series.

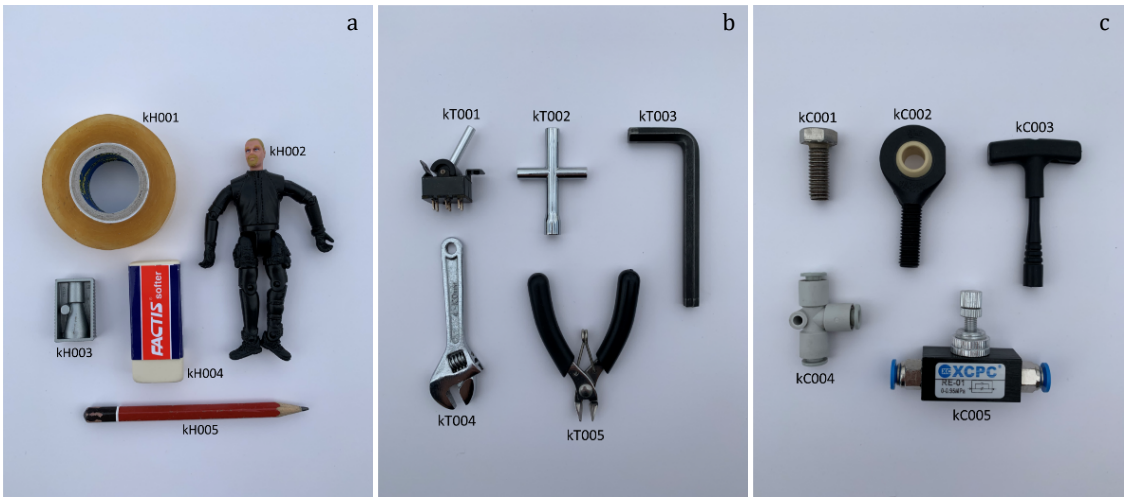


***Figure 5.1*. 15 objects within the known series. An object is considered known if it was included in any of the training datasets. (a)—known household items. (b)—known tool items. (c)—known component items.**

***Table 5.1*. Object register, known series.**

Series	Category	Object ID	Description
Known	Household class	kH001	Sellotape adhesive tape
		kH002	Figurine
		kH003	Pencil sharpener
		kH004	Factis eraser
		kH005	Pencil
	Tool class	kT001	Electrical switch
		kT002	Small cross wrench
		kT003	Large hex key
		kT004	Small adjustable wrench
		kT005	Small side cutters
	Component class	kC001	Small silver bolt
		kC002	Rod end bearing
		kC003	Pull start handle
		kC004	Pneumatic T-splitter
		kC005	XCPC pneumatic valve

The unknown series contains 45 unique objects. Objects within this series are labelled as unknown because they are not seen during training. As such, unknown objects are used exclusively for testing purposes. Figure 5.2 shows the objects within the unknown series and their related identifier. Table 5.2 is the associated object register.

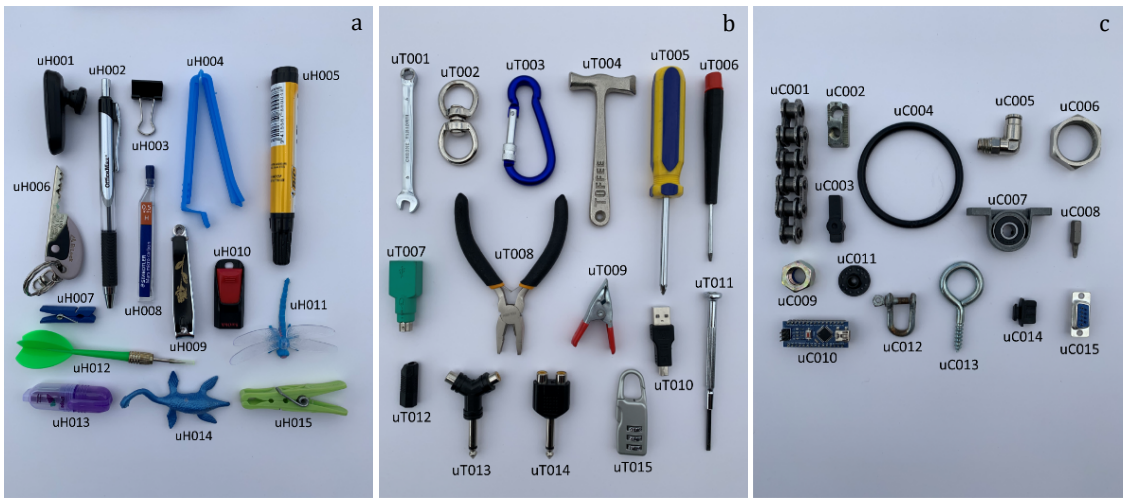


***Figure 5.2*. 45 objects that make up the unknown series. Objects within this series are not included in any training datasets. (a)—unknown household items. (b)—unknown tool items. (c)—unknown component items.**

***Table 5.2*. Object register, unknown series**

Series	Category	Object ID	Description
Unknown	Household class	uH001	Bluetooth earphone
		uH002	Black Officemax pen
		uH003	Small foldback clip
		uH004	Blue sealer
		uH005	Bic permanent marker
		uH006	Key
		uH007	Small blue peg
		uH008	Pencil refill
		uH009	Rose nail clipper
		uH010	Cruze glide USB

		uH011	Blue dragonfly toy
		uH012	Green dart
		uH013	Purple grape highlighter
		uH014	Blue Plesiosaur toy
		uH015	Light green clothes peg
	Tool class	uT001	Size 6 wrench
		uT002	Steel swivel
		uT003	Blue carabiner
		uT004	Walkers nutcracker
		uT005	Blue and yellow screwdriver
		uT006	Small red and black screwdriver
		uT007	Green USB converter
		uT008	Small pliers
		uT009	Red spring clip
		uT010	Black USB converter
		uT011	Silver flathead screwdriver
		uT012	Black power jack part
		uT013	Quarter inch audio Y-splitter
		uT014	Quarter inch audio T-splitter
		uT015	Silver combination lock
	Component class	uC001	Motorcycle chain
		uC002	Aluminium corner component
		uC003	Small black component
		uC004	Large rubber seal
		uC005	Silver corner pneumatic attachment
		uC006	Large pneumatic nut
		uC007	Shaft bearing
		uC008	Square driver socket component
		uC009	Large blue nylock nut
		uC010	Arduino Nano
		uC011	Circular black servo component
		uC012	Small D-clamp
		uC013	Silver eyelet screw
		uC014	Black rubber HDMI blank
		uC015	VGA connector

The miscellaneous series contains the remaining 40 objects. Objects within this set are also considered unknown, as they are not used to generate training data. This series includes objects that may be similar to objects within the two other series and are used for testing purposes exclusively. Note that the addition of such a third object series is common throughout literature and used to indicate a lower level of perceived variation. Objects within the miscellaneous series are shown in Figure 5.3. The object register for this series is tabulated in Table 5.3.

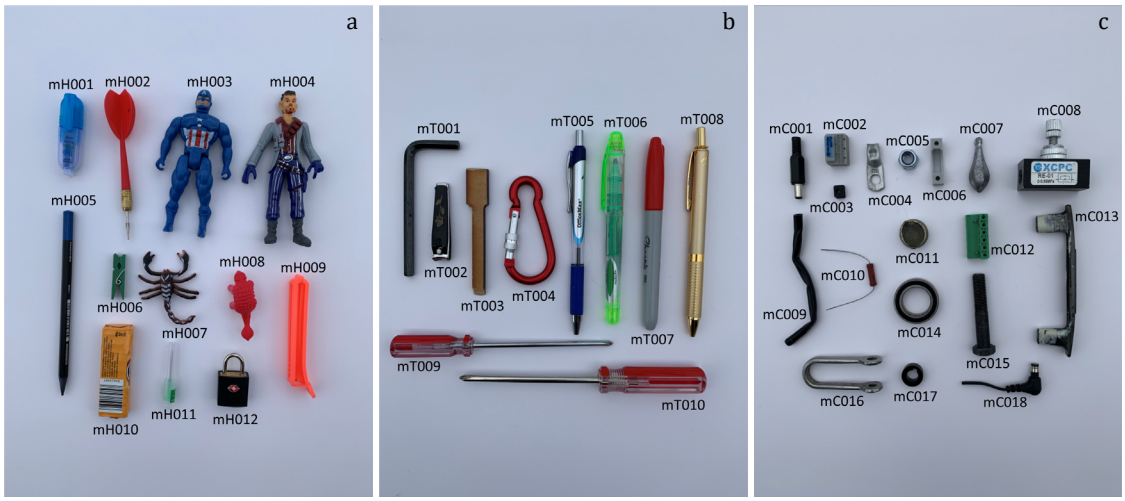


***Figure 5.3*. 40 objects included in the miscellaneous series. Objects within this series are not included in any training datasets. (a)—miscellaneous household items. (b)—miscellaneous tool items. (c)— miscellaneous component items.**

***Table 5.3*. Object register, miscellaneous series.**

Series	Category	Object ID	Description
Misc.	Household class	mH001	Blue blueberry highlighter
		mH002	Red dart
		mH003	Blue Captain America figurine
		mH004	Fixed figurine
		mH005	Black and blue pencil
		mH006	Small green peg
		mH007	Brown scorpion toy
		mH008	Red Euoplocephalus toy
		mH009	Red sealer
		mH010	PK chewing gum
		mH011	Unopened needle
		mH012	Small black key lock
	Tool class	mT001	Medium hex key
		mT002	Small nail clipper
		mT003	Wooden handle
		mT004	Red carabiner
		mT005	Blue Officemax pen
		mT006	Green Officemax pen
		mT007	Red Sharpie
		mT008	Golden pen
		mT009	Small red screwdriver
		mT010	Large red screwdriver
	Component class	mC001	Black power jack
		mC002	Grey USB cover
		mC003	Small black rubber insert
		mC004	Silver battery terminal
		mC005	Small nylock nut
		mC006	Grey 3D printed part
		mC007	1oz sinker
		mC008	XCPC pneumatic valve – no fittings
		mC009	Black pneumatic line
		mC010	Red resistor
		mC011	Water tap filter
		mC012	Green terminal black

	mC013	Door handle
	mC014	Ball bearing
	mC015	Long black bolt
	mC016	Steel U
	mC017	Black circular motor coupling
	mC018	Power jack with wire

Chapter 6

Stage 1: Object classification

6.1 Generating training data

Deep learning methodologies bring robustness at the cost of collecting large quantities of data. Amongst other factors, this robustness is tied to the number of samples used for training, i.e. more data generally leads to more robust predictions about new data. Data quality and the amount of useful information contained within this data, are key factors that determine how well an ML algorithm forms generalisations about the data, thereby affecting predictive power. To improve the quality of training data, the collection process employed in this thesis avoids human interaction where possible. Samples are taken from real sensor data during typical prototype operation, collected by haphazardly placing objects on the prototype conveyor belt. Detected objects are automatically moved to the vision system, where 1920×1080 RGB images of the presented object are captured, converted to greyscale, processed, segmented, cropped, scaled and saved within the appropriate class folder, which is specified by the human operator. Samples are generated automatically via the same process as obtaining I_{BB}. Section 4.3 provides further information related to the segmentation process. Figure 6.1 shows examples of typical samples included in the stage 1 dataset.

***Figure 6.1*. Examples of typical images of objects within the known series included in the stage 1 dataset. Object IDs from left to right, top to bottom: kH001, kH002, kH003, kH004, kH005, kT001, kT002, kT003, kT004, kT005, kC001, kC002, kC003, kC004, kC005, noObject.**

The stage 1 dataset contains training samples for 16 classes, generated exclusively from the 15 objects within the known object series. One class is assigned to each object within this series. For more information regarding the object test pool, see Chapter 5. Note that the 16th class relates to samples in which no objects are present. This class was added for redundancy, as it may be possible that there is some error which leads to no objects present within classification window I_{BB}, although, this error is largely avoided using traditional DIP techniques. During training, it was found that this additional class very slightly assisted in reducing inter-class confusion, resulting in increased classification accuracy, recall rates and precision rates. For more information regarding the training process, refer to Section 6.2. Normally, the proposed prototype reliably positions candidate objects at the centre of the vision system. However, to further improve the robustness of the stage 1 classifier, atypical samples were included in the training set. Such samples may vary from the ideal in terms of light conditions. Objects within these samples may also be significantly off-centre and/or arranged in challenging orientations. Partial views of objects were also incorporated

into the stage 1 dataset. It should be noted that care was taken to curate images of objects in various configurations and orientations. This confounds the classification task significantly for highly reconfigurable objects, e.g., figurine (kH002), electrical switch (kT001) and small adjustable wrench (kT004). Figure 6.2 illustrates some atypical examples of samples within the stage 1 training dataset.

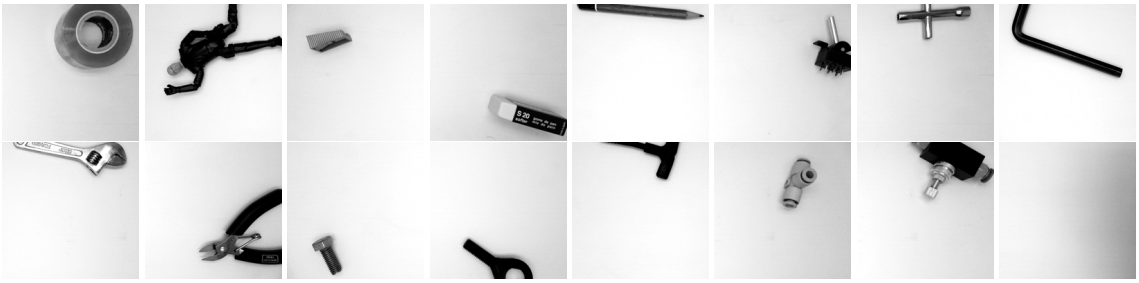


***Figure 6.2*. Extreme examples of atypical samples within the stage 1 dataset. Object IDs from left to right, top to bottom: kH001, kH002, kH003, kH004, kH005, kT001, kT002, kT003, kT004, kT005, kC001, kC002, kC003, kC004, kC005, noObject.**

16,000 RGB samples of size I_{BB} were captured. 1,000 unique windows were associated with each of the 15 objects in the known series and 1 blank class. These samples were downscaled to 198 × 198 and converted to greyscale, i.e. pixel intensity levels range from [0 . . 255]. Additional samples were generated from this set through Gaussian blurring, 90° rotation, 180° rotation and 270°rotation, yielding 4 extra samples per input sample, for a total of 4,000 generated samples per 1,000 original samples. Thus, the stage 1 dataset contains a total of 80,000 greyscale samples of size 198 × 198, split evenly between 16 classes, resulting in 5,000 samples per class. Figure 6.3 illustrates 4 additional samples generated from an input sample. Table 6.1 presents the stage 1 dataset (STG1_v4) register.

***Figure 6.3*. Illustration of additional training samples generated from an input sample. (a)—input sample. (b)—sample blurred by Gaussian filter. (c)—sample rotated by 90°. (d)—sample rotated by 180°. (e)—sample rotated by 270°.**

***Table 6.1*. Stage 1 dataset register (STG1_v4). Samples are greyscale. Sample size is set to 198 × 198.**

Class designation	Corresponding object ID	Description	Number samples
selloTapeObject	kH001	Sellotape adhesive tape	5,000
figurineObject	kH002	Figurine	5,000
pencilSharpenerObject	kH003	Pencil sharpener	5,000
eraserObject	kH004	Factis eraser	5,000
pencilObject	kH005	Pencil	5,000
switchObject	kT001	Electrical switch	5,000
crossWrenchObject	kT002	Small cross wrench	5,000
allenKeyObject	kT003	Large hex key	5,000
wrenchObject	kT004	Small adjustable wrench	5,000
wireCutterObject	kT005	Small side cutters	5,000
boltObject	kC001	Small silver bolt	5,000
rodEndBearingObject	kC002	Rod end bearing	5,000
pullStartObject	kC003	Pull start handle	5,000

TsplitterObject	kC004	Pneumatic T-splitter	5,000
flowValveObject	kC005	XCPC pneumatic valve	5,000
noObject	-	No objects present	5,000
			80,000

Many legacy datasets also exist. Notable datasets are reproduced in Tables 6.2-6.5. Table 6.2 contains a variant of the stage 1 dataset in which I_{BB} size and dimensionality are retained. Note that the noObject class is not present in this dataset.

***Table 6.2*. Variant stage 1 dataset register (STG1_v4_HDC_clip). Samples in colour. Sample size is set to 790 × 790.**

Class designation	Corresponding object ID	Description	Number samples
selloTapeObject	kH001	Sellotape adhesive tape	1,000
figurineObject	kH002	Figurine	1,000
pencilSharpenerObject	kH003	Pencil sharpener	1,000
eraserObject	kH004	Factis eraser	1,000
pencilObject	kH005	Pencil	1,000
switchObject	kT001	Electrical switch	1,000
crossWrenchObject	kT002	Small cross wrench	1,000
allenKeyObject	kT003	Large hex key	1,000
wrenchObject	kT004	Small adjustable wrench	1,000
wireCutterObject	kT005	Small side cutters	1,000
boltObject	kC001	Small silver bolt	1,000
rodEndBearingObject	kC002	Rod end bearing	1,000
pullStartObject	kC003	Pull start handle	1,000
TsplitterObject	kC004	Pneumatic T-splitter	1,000
flowValveObject	kC005	XCPC pneumatic valve	1,000
			15,000

Table 6.3 presents an early version of the stage 1 dataset (STG1_v3) used in an initial trial of the proposed methodology.

***Table 6.3*. Early stage 1 dataset register (STG1_v3). Samples are greyscale. Sample size is set to 198 × 198.**

Class designation	Corresponding object ID	Description	Number samples
pencilObject	kH005	Pencil	1,000
switchObject	kT001	Electrical switch	1,000
allenKeyObject	kT003	Large hex key	1,000
wrenchObject	kT004	Small adjustable wrench	1,000
wireCutterObject	kT005	Small side cutters	1,000
boltObject	kC001	Small silver bolt	1,000
rodEndBearingObject	kC002	Rod end bearing	1,000
pullStartObject	kC003	Pull start handle	1,000
TsplitterObject	kC004	Pneumatic T-splitter	1,000
flowValveObject	kC005	XCPC pneumatic valve	1,000
noObject	-	No objects present	1,000
			11,000

***Table 6.4*. Early stage 1 dataset register (STG1_v2). Samples in colour. Sample size is set to 790 × 790.**

Class designation	Corresponding object ID	Description	Number samples
pencilObject	kH005	Pencil	1,000
switchObject	kT001	Electrical switch	1,000
allenKeyObject	kT003	Large hex key	1,000
wrenchObject	kT004	Small adjustable wrench	1,000
wireCutterObject	kT005	Small side cutters	1,000
boltObject	kC001	Small silver bolt	1,000
rodEndBearingObject	kC002	Rod end bearing	1,000
pullStartObject	kC003	Pull start handle	1,000
TsplitterObject	kC004	Pneumatic T-splitter	1,000
flowValveObject	kC005	XCPC pneumatic valve	1,000
			10,000

***Table 6.5*. Legacy stage 1 dataset register (STG1_v1). Samples are colour. Sample size is set to 790 × 790.**

Class designation	Corresponding object ID	Description	Number samples
bolt	kC001	Small silver bolt	250
ellenKey	kT003	Large hex key	250
screwDriver	-	Small screwdriver	250
			750

6.2 Learning to classify

When learning a new classification task, it is crucial to balance model complexity with classification difficulty. Generally, the difference in training accuracy and validation accuracy is used to gauge the variance of a model. Common strategies to avoid high variance, or overfitting, include a reduction in model complexity, a reduction in the dimensionality of the training data or the collection of more training data. These strategies are supported by contemporary ML theory [135, 206]. Although many underlying principles influence the behaviour of an ML algorithm, the construction of a new dataset and training new models largely rely on trial-and-error and expertise. To this end, 81 tentative networks were initially trained with variations in learning rate, batch size, model architecture, dataset construction, data dimensionality, training iterations and momentum optimisers. The subsequent dataset, denoted as STG1_v3 (Table 6.3, Section 6.1), consisted of 11,000, 198 × 198 greyscale samples pertaining to 10 objects within the known object series and 1 blank class, resulting in 11-class classification. The resulting model, *97classNet_V3_11*, served as a baseline for further refinement. This model was trained for 20 epochs with a mini-batch size of 10 and initial learning rate of 5×10^{-4}. 80% of the samples in the STG1_v3 dataset were used for training. The remaining 20% were used for validation. Samples were split between these two subsets randomly but maintained an even class distribution. *97classNet_V3_11* correctly classified 97% of the validation set. Figure 6.4 illustrates the architecture of this CNN.

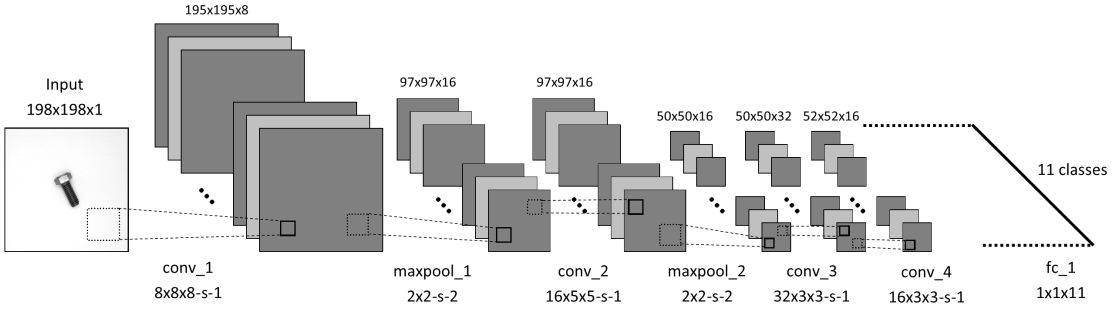


***Figure 6.4. 97classNet_V3_11* network architecture. The first convolutional layer filters the 198 × 198 × 1 input image with 8 kernels of size 8 × 8 and a stride of 1 pixel. The second convolution takes as input the response-normalised and pooled output of the first convolutional layer, filtered with 16 kernels of size 5 × 5 for each of the resultant input layers. The third convolutional layer has 32 kernels of size 3 × 3, connected to the normalised and pooled previous convolution. The final convolutional layer has 16 kernels of size 3 × 3. The fully connected layer has 11 neurons, one for each output class. In total, this network has 484,851 trainable parameters.**

At this stage in development, downscaling and greyscale conversion had been anecdotally demonstrated to improve performance, in accordance with ML theory. Here, performance relates to several desired criteria, e.g., model accuracy, computation time, training data memory usage, training time, network memory usage, etc. To empirically investigate the utility and consequent impact of such pre-processing techniques, 6 identical networks were trained with 6 variants of the same training dataset, i.e. the STG1_v4_HDC_clip dataset and an additional 5 variations of this set. Note that samples within these datasets simply vary in size and dimensionality. STG1_v4_HDC_clip consists of 15,000 samples of the 15 objects within the known series. The 16th blank class is not included in this dataset. Table 6.2, Chapter 6.1 provides additional details. Table 6.6 presents the labels assigned to each of the corresponding dataset variations.

***Table 6.6.* Datasets used to investigate the impact of dimensionality and scale.**

Dimensionality	Sample size		
	790 × 790	395 × 395	198 × 198
RGB	STG1_v4_HDC_clip	STG1_v4_MDC_clip	STG1_v4_LDC_clip
Greyscale	STG1_v4_HDG_clip	STG1_v4_MDG_clip	STG1_v4_LDG_clip

The model used to investigate the utility of pre-processing is denoted as *classTestNet_V4_15*. The architecture of this model is illustrated in Figure 6.5. Note that although the network structure is consistent between datasets, the resulting number of input layers to each convolution may differ due to differences in dimensionality and scale of the respective inputs.

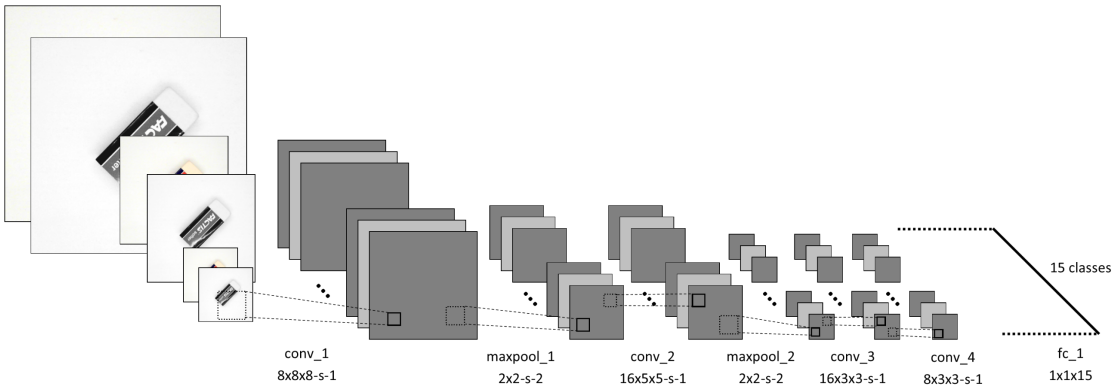


***Figure 6.5. classTestNet_V4_15* network architecture. The first convolutional layer filters the input image with 8 kernels of size 8 × 8 and a stride of 1 pixel. The second convolution takes as input the response-normalised and pooled output of the first convolutional layer, filtered with 16 kernels of size 5 × 5 for each of the resultant input layers. The third convolutional layer has 16 kernels of size 3 × 3, connected to the normalised and pooled previous convolution. The final convolutional layer has 8 kernels of size 3 × 3. The fully connected layer has 15 neurons, one for each output class.**

6 variations of *classTestNet_V4_15* were trained. Each network was trained for 5 epochs with a mini-batch size of 40, for a total of 1,310 training iterations. The initial learning rate was set to 1×10^{-4}. Of the samples in the corresponding dataset, 70% were used for training, with the remaining 30% used for validation. Training curves for these 6 models are shown in Figure 6.6.

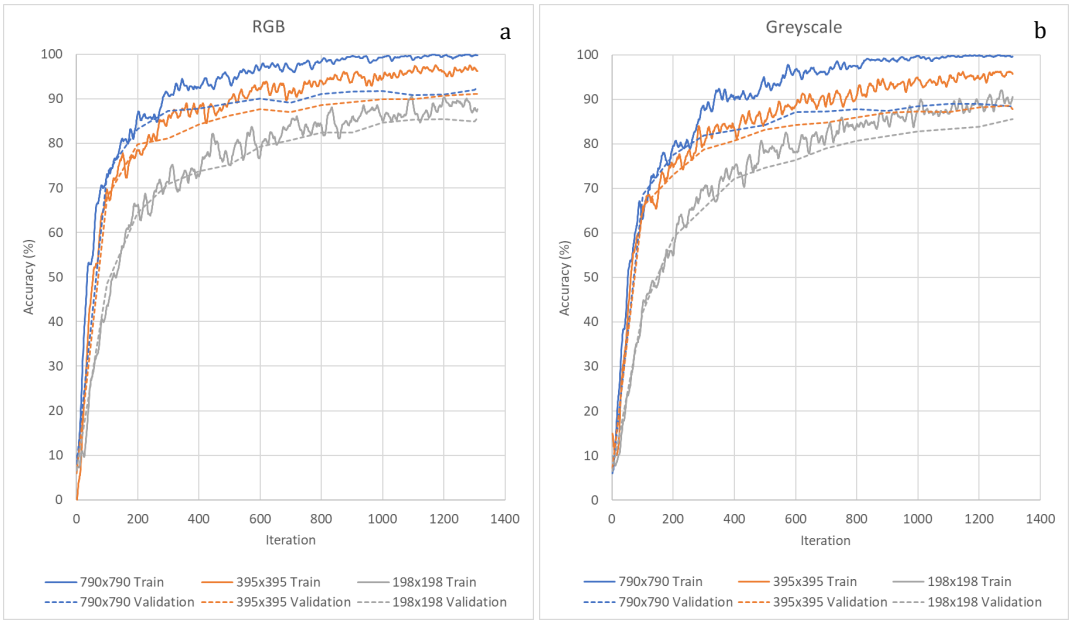


***Figure 6.6.* Training curves for 6 identical networks trained with varying datasets. (a)—*classTestNet_V4_15* trained with colour data at scales: 790 × 790, 395 × 395 and 198 × 198. (b)—*classTestNet_V4_15* trained with greyscale data at scales: 790 × 790, 395 × 395 and 198 × 198.**

From the training curves in Figure 6.6, it is apparent that all 6 models are overfitting to some degree, i.e. training accuracy is significantly higher than validation accuracy. This is somewhat supported by Figure 6.7, which illustrates the final training accuracy, validation accuracy and the resulting difference.

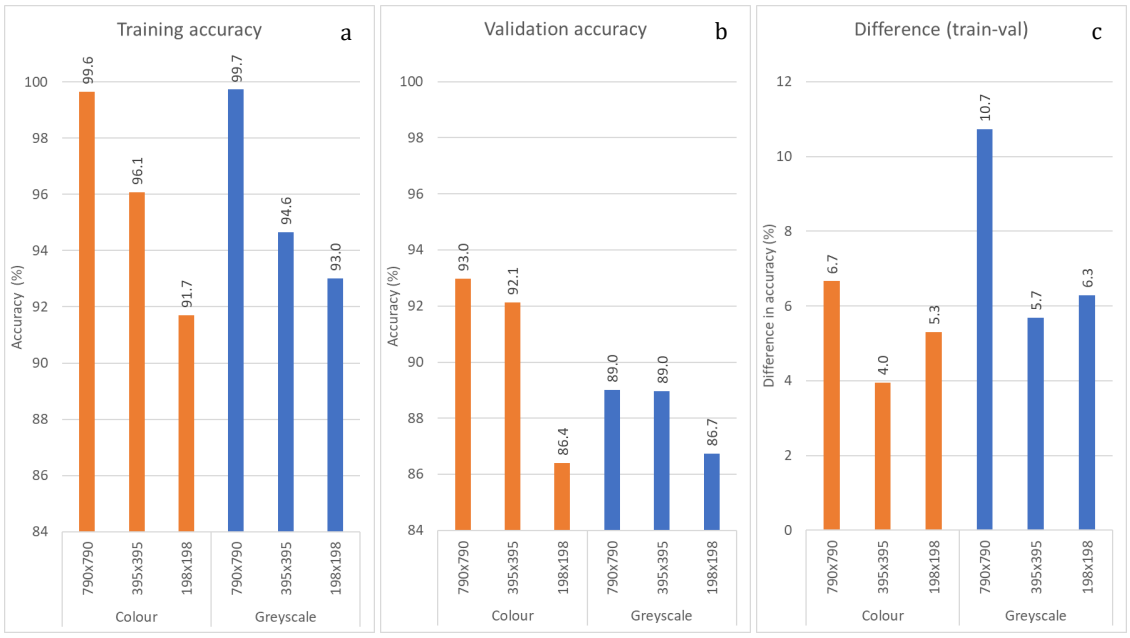


***Figure 6.7*. (a)—training accuracy of *classTestNet_V4_15* after 1,310 training iterations with variations in training data. (b)—resulting validation accuracy of *classTestNet_V4_15*. (c)—difference in training and validation accuracy.**

Validation data is separated from training data to gauge the generalised performance of the trained model on new data. Ideally, the difference in validation accuracy and training accuracy should be minimal. The disparity between training and validation data shown in Figure 6.7—c may reflect a lack of variety in training data, insufficient training data, inappropriate training properties or the model may be too complex for the classification task. Note that the model architecture of these networks is based on *97classNet_V3_11*, a network developed to classify 198 × 198, greyscale samples. The training curves shown in Figure 6.6 also illustrate the difference in difficulty of the classification task associated with each respective dataset. Greyscale data at 198 × 198 for instance, contains significantly less information, therefore confounding the classification task, reflected in the learning rate of the associated network. This may suggest that a significantly less complex network could be used for classifying colour data at 790 × 790. The training accuracy of colour models at 395 × 395 and 198 × 198 and greyscale models at 395 × 395 and 198 × 198 appear to be trending upward in Figure 6.6. This suggests that better classification accuracy could be achieved with longer training periods, although, this may not improve validation accuracy. The network trained with 198 × 198, colour data appears to overfit the least.

Network architecture and training properties are flexible and can be refined to achieve the desired accuracy once an appropriate training dataset has been established. At this stage of development, other associated metrics were also assessed. The time to classify a sample for instance, was significantly higher for colour images at scale 790 × 790. For colour images, classification time was 16.3 ms at 790 × 790, 13.6 ms at 395 × 395 and 8.0 ms at 198 × 198. Classification time decreased by half between images of size 790 × 790 and 198 × 198. Note that the time taken to scale these samples must be considered because this scaling process is not required for input size 790 × 790. Although this is the case, the time to scale is insignificant compared to classification time at larger scales (Figure 6.8—b). For greyscale samples, classification time was 9.7 ms at 790 × 790, 6.8 ms at 395 × 395 and 3.6 ms at 198 × 198. Greyscale conversion is required for preceding DIP processes and therefore does not need to be factored into classification time. At scale 790 × 790, greyscale images were classified in roughly half the time compared to 790 × 790 colour images. See Figure

6.8 for more details. The specifications of the computer hardware used to train and assess networks is provided in Section 9.2.

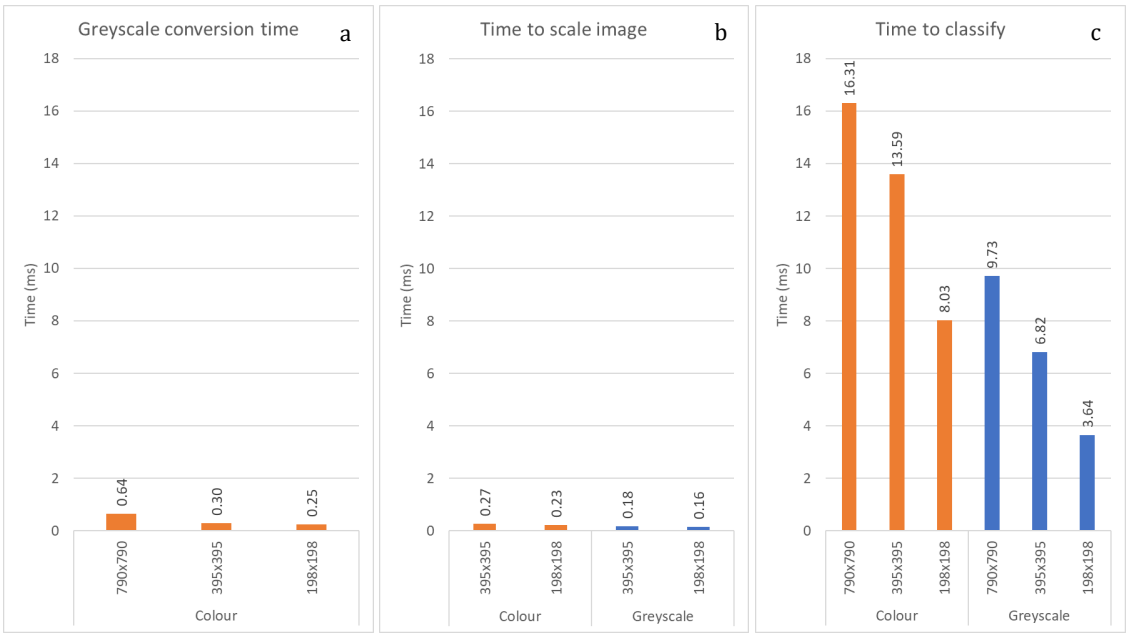


Figure 6.8. **(a)—greyscale conversion times for colour images at scales: 790 × 790, 395 × 395 and 198 × 198. (b)—time taken to rescale greyscale and colour samples of size 790 × 790 to 395 × 395 and 198 × 198. (c)—classification times for colour and greyscale-based networks at scales: 790 × 790, 395 × 395 and 198 × 198.**

Several other properties were also considered prior to settling on the proposed stage 1 dataset (STG1_v4). The disk space required to store training data for instance, was significantly higher for colour images. At scale 790 × 790 for example, colour images required more than double the space of their greyscale counterpart. Dataset size could be reduced 36-fold through greyscale conversion and downscaling. The amount of disk space occupied by a dataset is significant, as it affects read/write times during training. Moreover, it may not be practical to store and transfer large amounts of data. Training times were also significantly affected, with colour-based networks of size 790 × 790 requiring significantly more time to train, despite the fixed number of training iterations between training modalities. It took approximately 46 minutes to train colour images of size 790 × 790 for 1,310 iterations, compared to 4 minutes for 198 × 198 colour images trained for the same number of iterations. Training time was less than 3 minutes for the network trained with 198 × 198 greyscale images. This metric is particularly important for timely research, as many networks need to be trained before settling on an appropriate network architecture, dataset, training properties, etc. Figure 6.9 illustrates the resulting training dataset size, training time, network size and total number of trainable parameters for the *classTestNet_V4_15* network, trained with samples that vary in dimensionality and size.

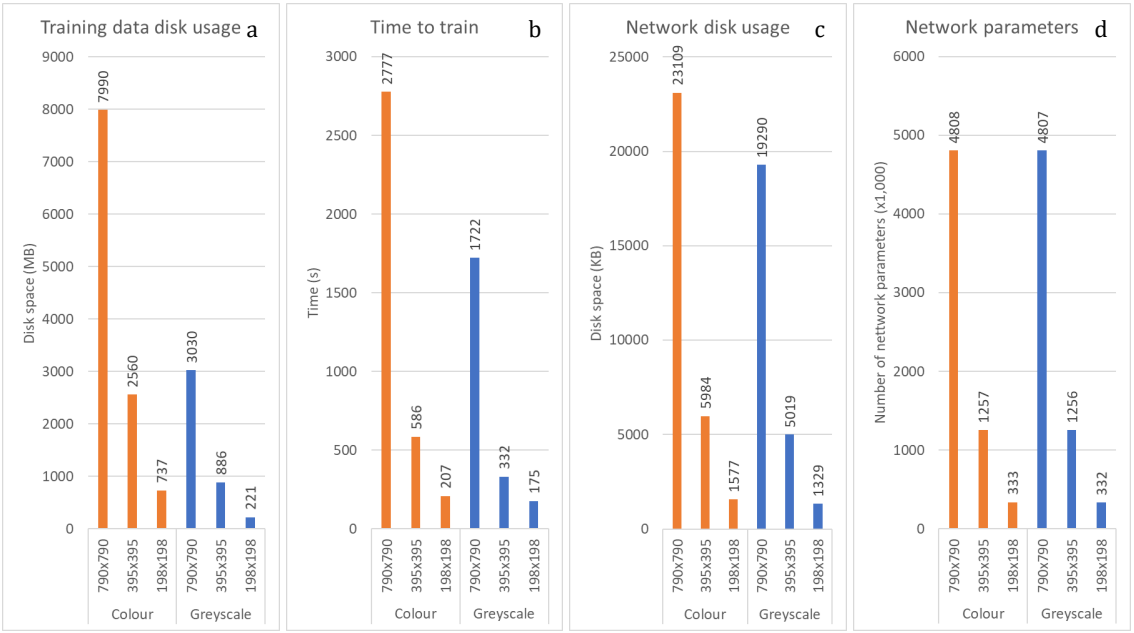


***Figure 6.9*. (a)—disk space occupied by datasets that vary in dimensionality and scale. (b)—time taken to train *classTestNet_V4_15* with various input layer sizes for 1,310 iterations. (c)—resulting disk space required to store variations of *classTestNet_V4_15*. (d)—resulting number of trainable parameters of *classTestNet_V4_15* trained with various input sizes.**

Although the inclusion of the 16th blank class, noObject, within the stage 1 dataset (STG1_v4) can be justified from the perspective of additional redundancy, networks trained with this additional class were found to have marginally reduced inter-class confusion, resulting in slightly increased classification accuracy, recall rates and precision rates. To illustrate, a variant of *97classNet_V3_11* with a fully connected layer with 16 neurons was trained using the STG1_v4 dataset. This dataset contains 80,000, 198 × 198, greyscale samples of 16 classes related to the known object series (Table 6.1, Section 6.1). In addition, an identical network with a fully connected layer with 15 neurons was trained using a variant of STG1_v4, in which the blank class had been removed. This dataset contained 75,000 198 × 198, greyscale samples of 15 classes. Each network was trained for 10 epochs, with a mini-batch size of 50, for a total of 11,200 iterations. The initial learning rate was set to 1×10^{-4}. Figure 6.10 and Figure 6.11 show the resulting confusion matrices of an identical network trained with and without the blank class included, respectively.

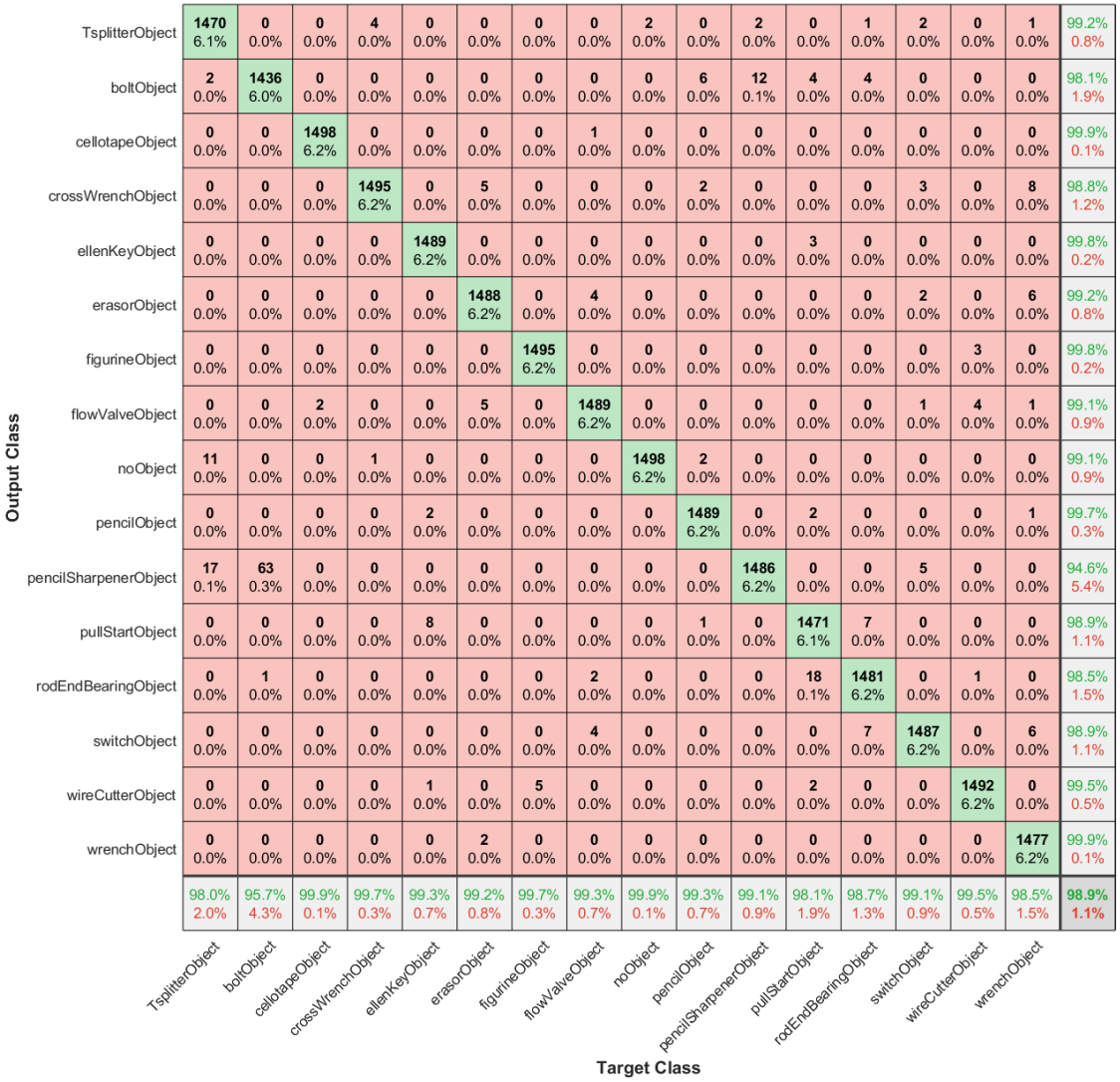


***Figure 6.10*. Confusion matrix of a *97classNet_V3_11*-based network with 16 output neurons trained with a 16th blank class.**

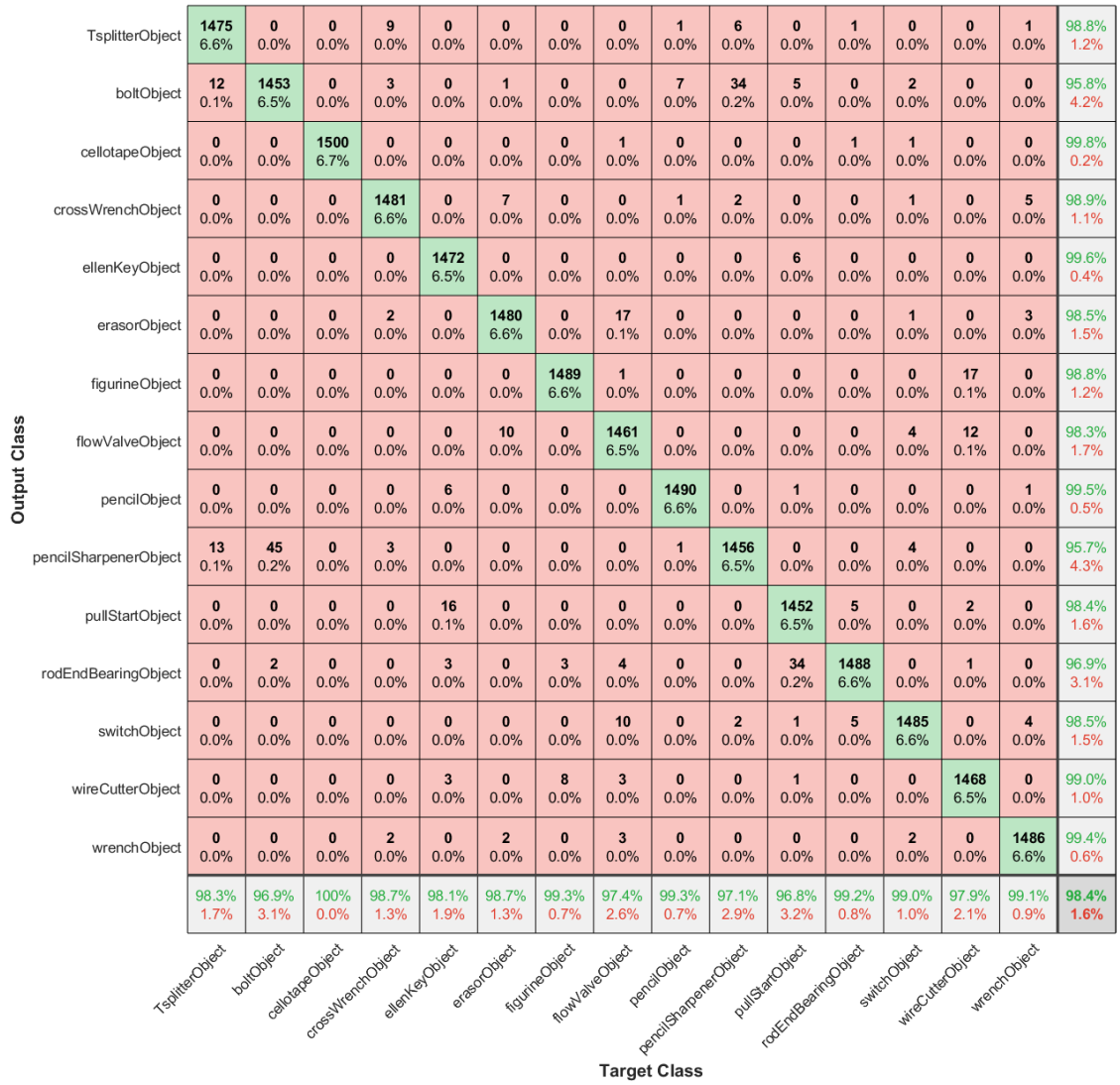


***Figure 6.11*. Confusion matrix of a *97classNet_V3_11*-based network with 15 output neurons trained without a 16th blank class.**

30% of each respective dataset was reserved for validation. The confusion matrices in Figure 6.10 and Figure 6.11 are calculated based on the classification rates of the validation subset. Training with the blank class increased validation accuracy by roughly 0.5%. Moreover, inter-class confusion in terms of class-wise precision and recall rates were generally improved with the inclusion of the blank class. Precision relates to the proportion of relevant predictions among all predictions, while recall relates to the total amount of relevant predictions. A comprehensive description of recall and precision is beyond the scope of this research. For more information regarding the relationship between these properties, refer to Buckland and Gey [386]. Fawcett provides a good introduction to the analysis of confusion matrices [387]. Although these rates are only slightly higher in the network trained with 16 output classes, it should be noted that the dataset with the blank class contains an additional 5,000 samples. Therefore, the slight increase in performance of the *97classNet_V3_11*-based network with 16 output neurons compared to the 15-neuron variant is noteworthy.

At this stage in development, it had been established that greyscale conversion, scale reduction and the inclusion of a blank class improved the practicality and performance of

ML algorithms employed for classification for the desired task. To test the effect of dataset size and training time, a network was trained with 2, 5, 10, 20, 50 and 70% of the data contained within the STG1_v4 dataset for 2, 5, 10, 15 and 30 epochs. For the number of samples corresponding to these percentages, refer to Table 6.7. A fixed 30% of the STG1_v4 dataset was used for validation, resulting in a validation set size of 24,000. Note that all networks trained in this context were validated using identical data. The network architecture, *classTestNet_V4_16,* used for this investigation is shown in Figure 6.12. Networks were trained with a mini-batch size of 30 and initial learning rate of 5×10^{-5}.

***Table 6.7*. Number of samples within STG1_v4 dataset corresponding to percentage. The total number of samples within this dataset is 80,000.**

Dataset percentage (%)	Number samples
2	1,600
5	4,000
10	8,000
20	16,000
50	40,000
70	56,000

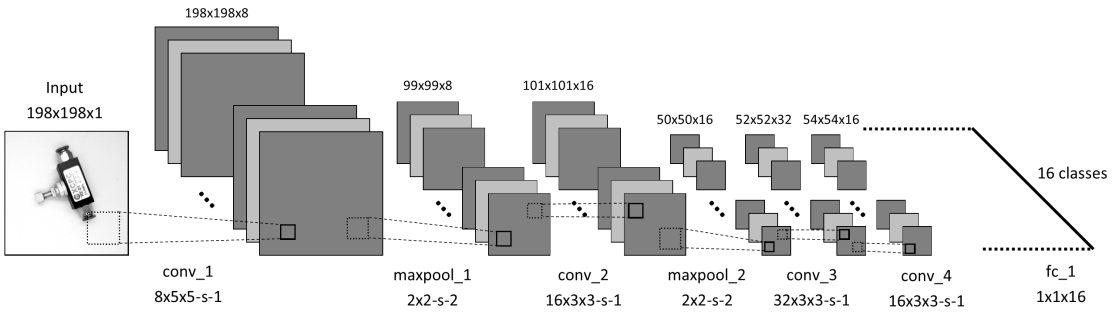


***Figure 6.12*. *classTestNet_V4_16* network architecture. The first convolutional layer filters the input image with 8 kernels of size 5×5 and a stride of 1 pixel. The second convolution takes as input the response-normalised and pooled output of the first convolutional layer, filtered with 16 kernels of size 3×3 for each of the resultant input layers. The third convolutional layer has 32 kernels of size 3×3, connected to the normalised and pooled previous convolution. The final convolutional layer has 16 kernels of size 3×3. The fully connected layer has 16 neurons, one for each output class. In total, this network has 757, 168 trainable parameters.**

The resulting training and validation curves for *classTestNet_V4_16* trained for 2 epochs with 2, 5, 10, 20, 50 and 70% of the data within the STG1_v4 dataset is shown in Figure 6.13. Similarly, networks trained with these variations in training sizes for 5, 10, 15 and 30 epochs are shown in Figure 6.14, Figure 6.15, Figure 6.16 and Figure 6.17, respectively.

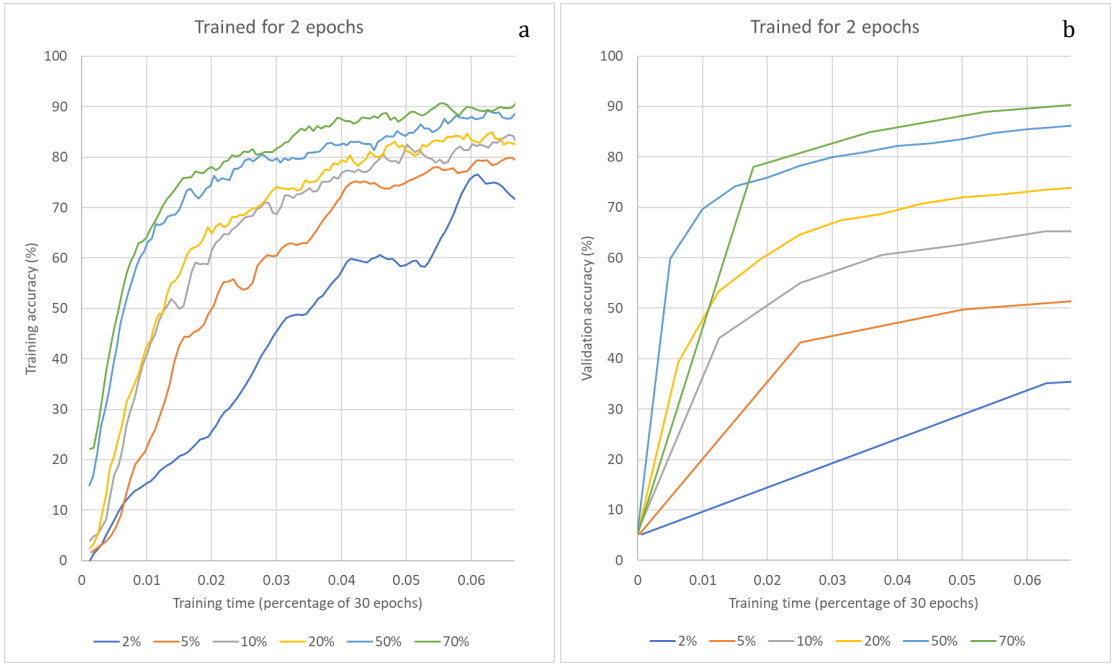


***Figure 6.13*. (a)—training curve of *classTestNet_V4_16*, trained for 2 epochs with 2, 5, 10, 20, 50 and 70% of the STG1_v4 dataset. (b)—validation curve of *classTestNet_V4_16*, trained for 2 epochs with 2, 5, 10, 20, 50 and 70% of the STG1_v4 dataset.**

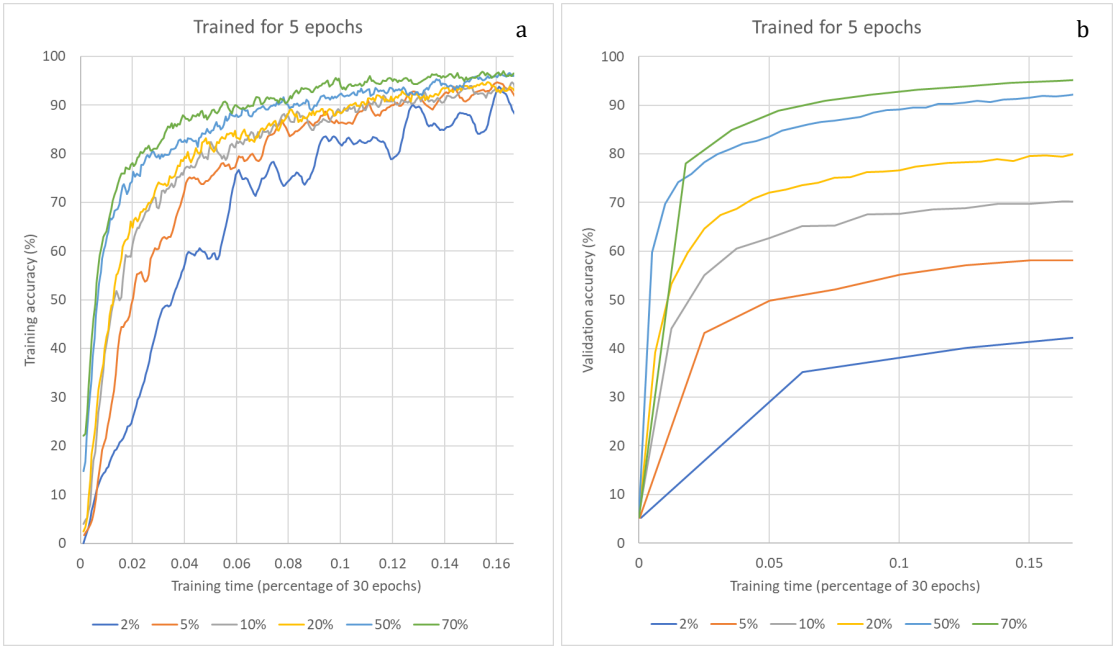


***Figure 6.14*. (a)—training curve of *classTestNet_V4_16*, trained for 5 epochs with 2, 5, 10, 20, 50 and 70% of the STG1_v4 dataset. (b)—validation curve of *classTestNet_V4_16*, trained for 5 epochs with 2, 5, 10, 20, 50 and 70% of the STG1_v4 dataset.**

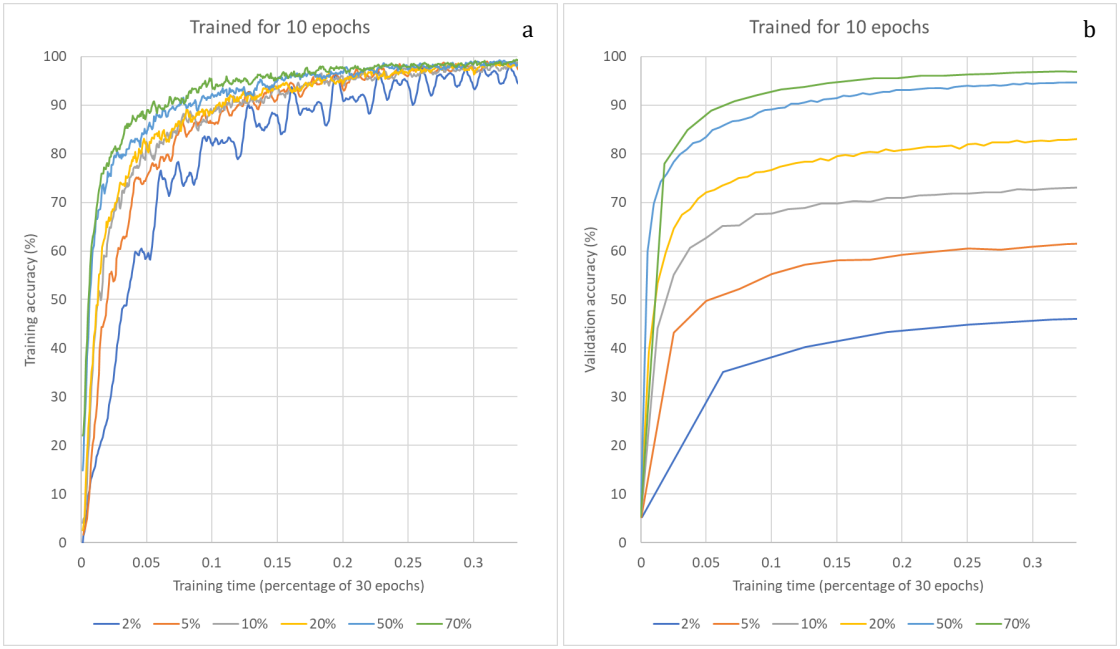


***Figure 6.15*. (a)—training curve of *classTestNet_V4_16*, trained for 10 epochs with 2, 5, 10, 20, 50 and 70% of the STG1_v4 dataset. (b)—validation curve of *classTestNet_V4_16*, trained for 10 epochs with 2, 5, 10, 20, 50 and 70% of the STG1_v4 dataset.**

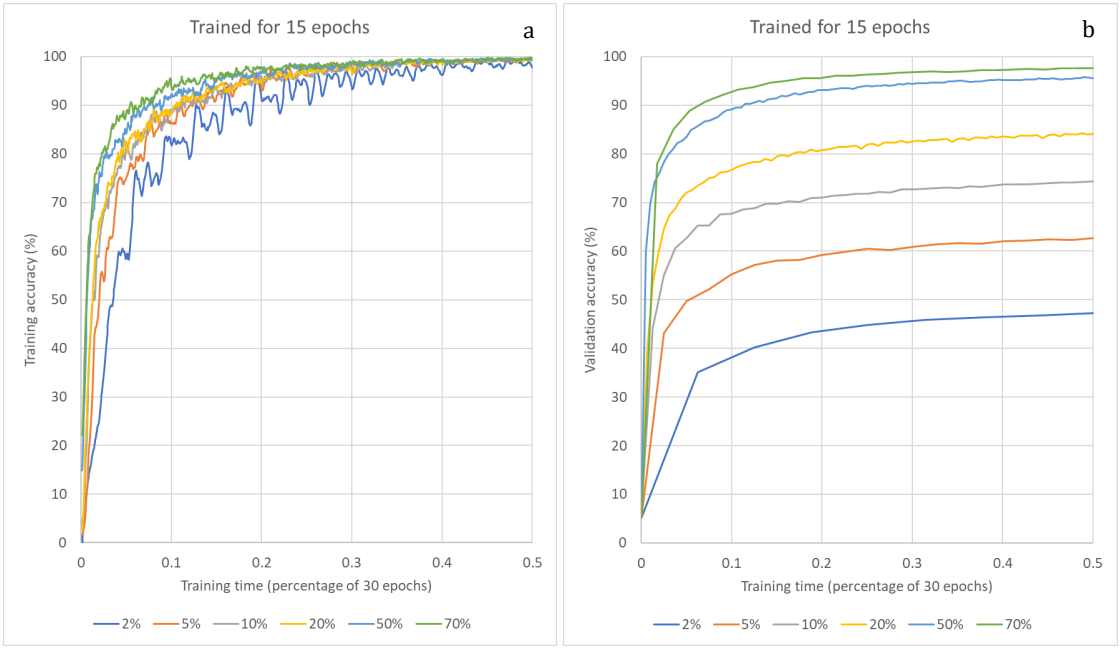


***Figure 6.16*. (a)—training curve of *classTestNet_V4_16*, trained for 15 epochs with 2, 5, 10, 20, 50 and 70% of the STG1_v4 dataset. (b)—validation curve of *classTestNet_V4_16*, trained for 15 epochs with 2, 5, 10, 20, 50 and 70% of the STG1_v4 dataset.**

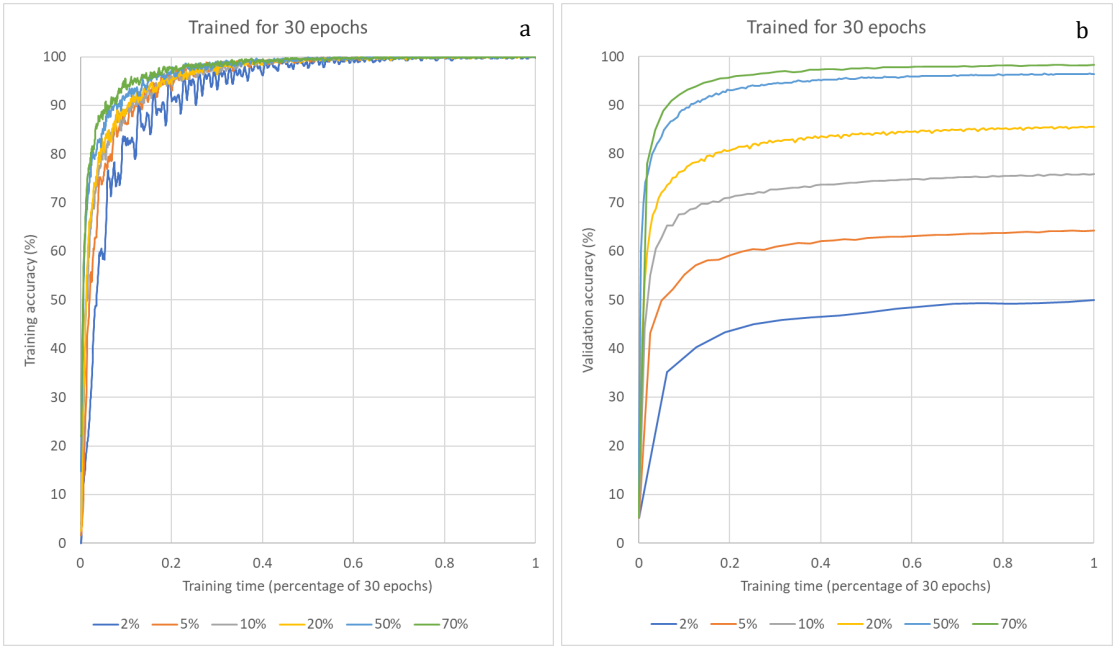


***Figure 6.17*. (a)—training curve of *classTestNet_V4_16*, trained for 30 epochs with 2, 5, 10, 20, 50 and 70% of the STG1_v4 dataset. (b)—validation curve of *classTestNet_V4_16*, trained for 30 epochs with 2, 5, 10, 20, 50 and 70% of the STG1_v4 dataset.**

Training noise fluctuated most significantly when training *classTestNet_V4_16* with only 2% of the STG1_v4 dataset and least significantly when trained with 70% of the dataset. This is expected, as it is more difficult to form widely applicable generalisations about data with only 1,600 examples (2% dataset). In contrast, 56,000 samples (70% dataset) offer significantly more variety. Larger sample pools tend to produce algorithms that generalise better to new data. This is reflected in the validation curves throughout Figures 6.13-6.17. Networks trained with less data tended to produce larger disparities between training accuracy and validation accuracy. This is particularly evident in Figure 6.20—which illustrates the difference between final training and validation accuracy achieved by the respective network. Figure 6.18 tabulates the final training accuracy of *classTestNet_V4_16* trained in various ways. Figure 6.19 tabulates the associated validation accuracy.

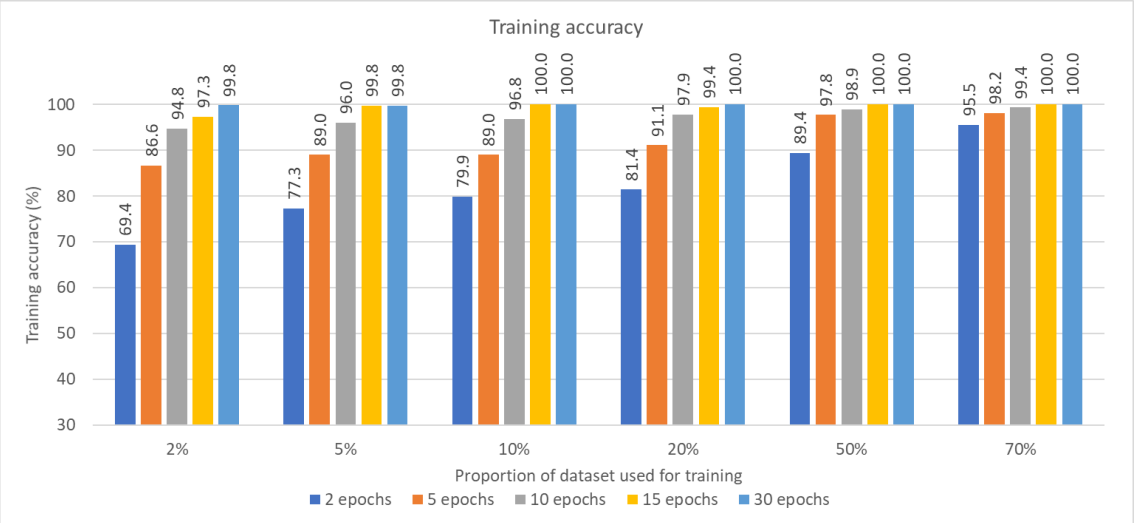


***Figure 6.18*. Training accuracy of *classTestNet_V4_16* trained for 2, 5, 10, 15 and 30 epochs with 2, 5, 10, 20, 50 and 70% training data.**

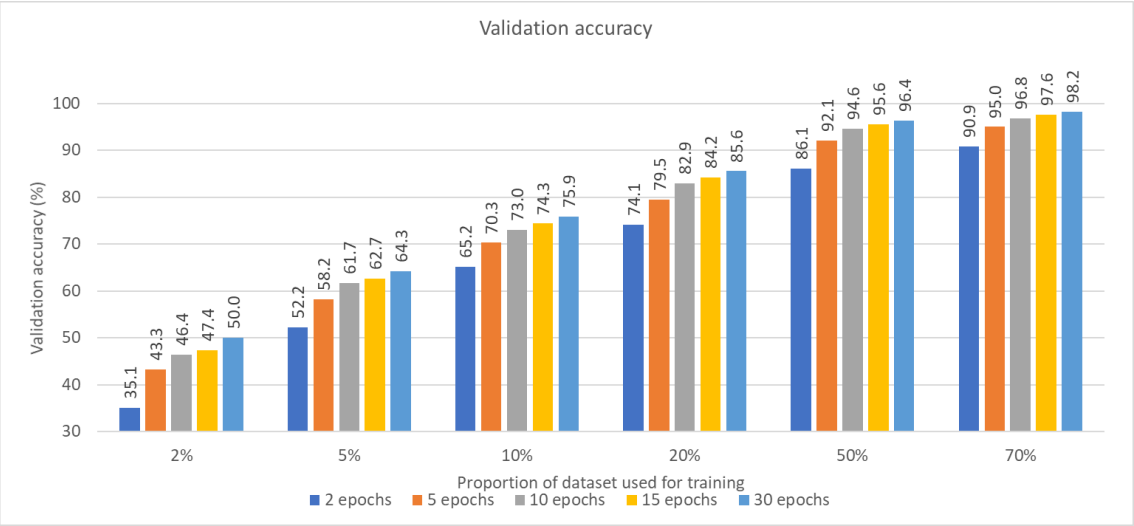


***Figure 6.19*. Validation accuracy of *classTestNet_V4_16* trained for 2, 5, 10, 15 and 30 epochs with 2, 5, 10, 20, 50 and 70% training data.**

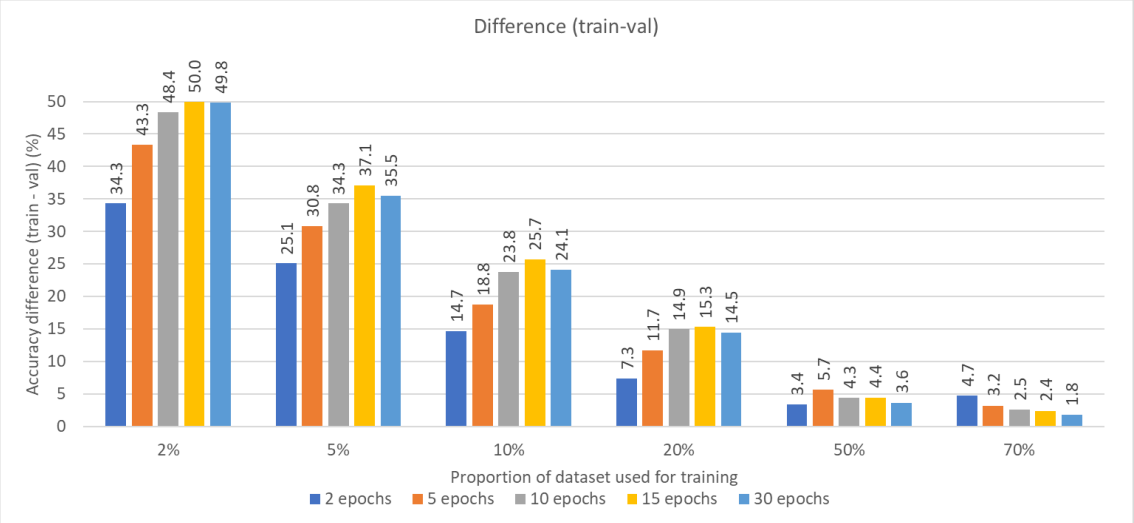


***Figure 6.20*. The absolute difference between training and validation accuracy of *classTestNet_V4_16* trained for 2, 5, 10, 15 and 30 epochs with 2, 5, 10, 20, 50 and 70% training data.**

Classifying a separate validation set provides a good indicator of network performance on new data. Figure 6.20 suggests that networks tend to overfit less when more data is available. This is consistent with ML theory. Training time can also affect network variance. Network performance for instance, tended to saturate when trained for around 10-15 epochs. This is evident in Figure 6.18, as training accuracy tended to stagnate or increase very slowly when trained for 10, 15 or 30 epochs, with 10% or more data. This is supported by Figure 6.17, where training accuracy seems to converge after 12 epochs for networks trained with 10% or more data. This stagnation suggests that the corresponding network has learned features that generalise very well to the training set. However, these features may not be useful when making predictions about new data. Even when trained for 30 epochs, 2% training data only resulted in the correct classification of 50% of the validation set (Figure 6.19), while correctly classifying 99.8% of the training data (Figure 6.18). Conversely, 90.9% validation accuracy is achieved after only 2 epochs when 70% of the STG1_v4 data was used during training. Moreover, *classTestNet_V4_16* shows little to no increase in training accuracy when trained for 30 epochs—regardless of the amount of data

used (Figure 6.18). However, the generalised performance on validation data increased logarithmically based on the amount of data utilised during training.

In total, over 300 networks were trained prior to settling on a network architecture suitable for the stage 1 classification task, aiming to optimise computational efficiency, dataset accuracy and practical feasibility. The final architecture of this network, denoted *classNet_V4_16*, is identical to the architecture of *classTestNet_V4_16*, illustrated in Figure 6.12. *classNet_V4_16* was trained for 15 epochs, with a mini-batch size of 50, resulting in 19,200 iterations. The initial learning rate was set to 5×10^{-5}. Training and validation data were taken from the STG1_V4 dataset. 56,000 (70%) images were used for training. The remaining 24,000 (30%) were used for validation. Samples were split randomly, but class distribution was maintained. It took approximately 44 minutes to train this network. This network occupied 2,877 KB of disk space. The total number of trainable parameters were 757,168. Time to classify a sample was approximately 3.6 ms. After training, this network correctly classified 99.84% of the training set and 99.51% of the validation set. Figure 6.21 shows the training and loss curves for this network. Note that the training figures cited are only estimates and may vary even if all the training properties are consistent. This is because the training and validation sets are selected randomly. Moreover, validation accuracy only provides an estimate of real-world performance, in that a finite number of samples are assessed.

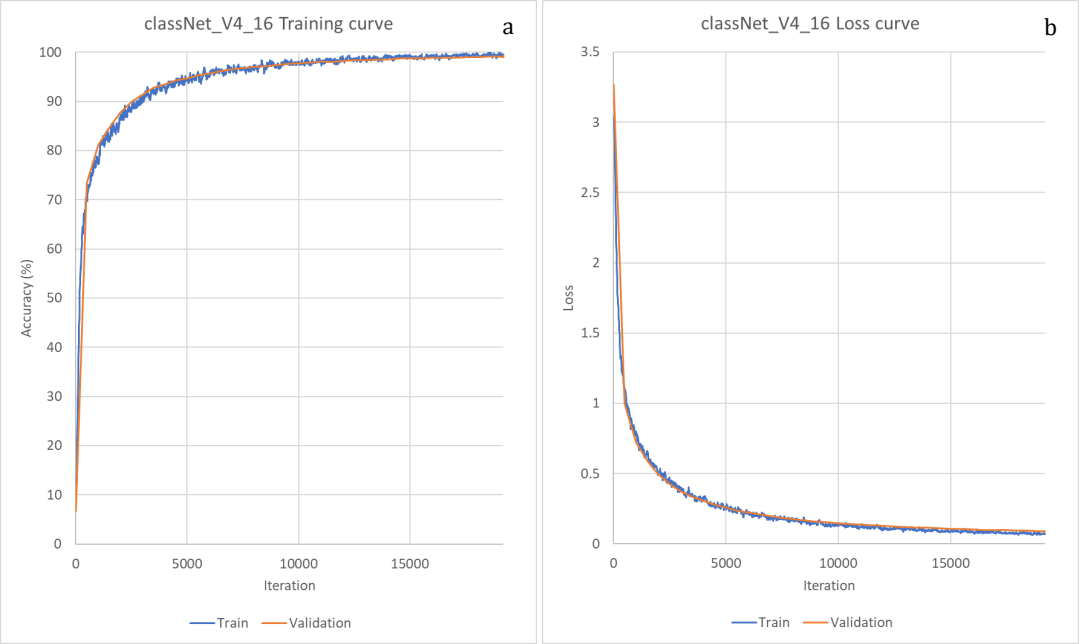


***Figure 6.21*. (a)—training and validation curves of *classNet_V4_16*, trained for 15 epochs with 70% of the STG1_v4 dataset. (b)—the corresponding loss curve of *classNet_V4_16*.**

The high validation performance of *classNet_V4_16* suggests that the features learned during training generalise well to new data. The small disparity (0.33%) between training and validation accuracy suggests that model complexity is well-suited to the difficulty of the classification task. Note that *classNet_V4_16* performs better on the validation set throughout the early stages of training. This is likely caused by dropout. By disabling some neurons during training, information about each sample is lost, resulting in subsequent layers classifying the sample based on incomplete input representation, forcing the network to learn redundant representations of the data. This is ideal during training, as the network

cannot rely on the activation of any set of hidden units since they may be disabled at any time during training. Raschka and Mirjalili suggest that dropout produces learned features that are more general and robust [206]. During validation, all units are available to the network, thus slightly better performance may be achieved on the validation set during the early stages of training.

A mini-batch size of 50 was found to increase validation performance and helped to smooth the training curve. Larger batch sizes could not be implemented due to memory constraints imposed by the GPU. A training time of 15 epochs was ideal to avoid overfitting. The confusion matrix for *classNet_V4_16* on the validation set is shown in Figure 6.22.

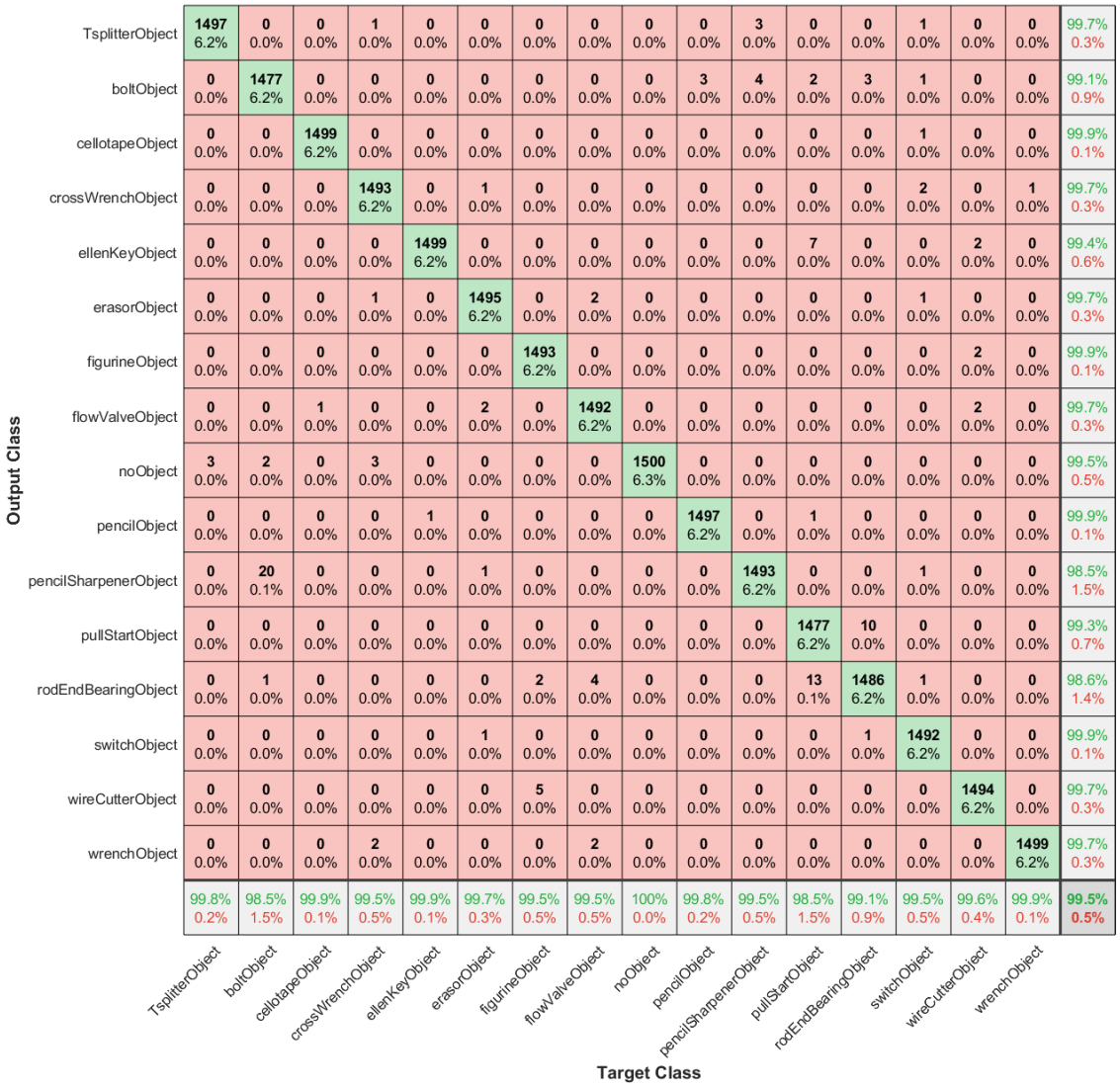


***Figure 6.22*. Confusion matrix of the *classNet_V4_16* network, calculated using validation data.**

Visualising the activations of a CNN may offer some understanding of the hierarchical representations learned by a network and which areas of the sample are given attention. Earlier layers tend to learn simple features such as edges and colour, while deeper layers learn complex features like shape. For an excellent discussion related to the understanding of network visualisation, the reader is directed to the Google Brain affiliated article by Carter *et al.* [388]. The activations for *classNet_V4_16*, given the sample shown in Figure 6.23, are visualised in Figures 6.24-6.27.

***Figure 6.23*. Input sample used to illustrate the activations of *classNet_V4_16*. Softmax function output: 0.9998, switchObject class (kT001).**

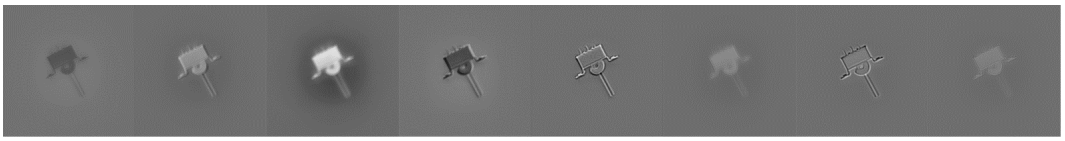


***Figure 6.24*. Visualisations of the conv_1 layer of *classNet_V4_16*, given an input sample of object kT001.**

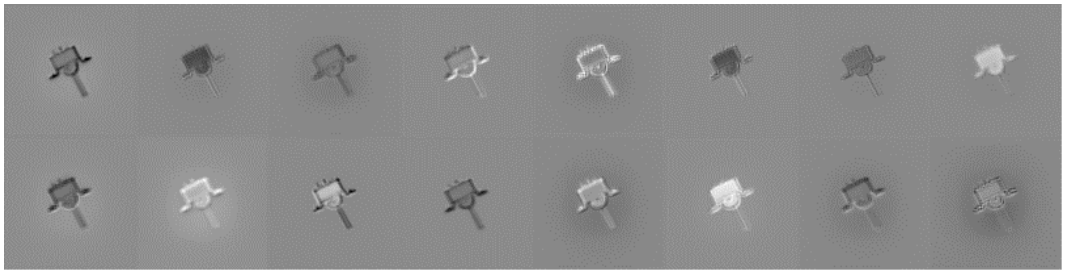


***Figure 6.25*. Visualisations of the conv_2 layer of *classNet_V4_16*, given an input sample of object kT001.**

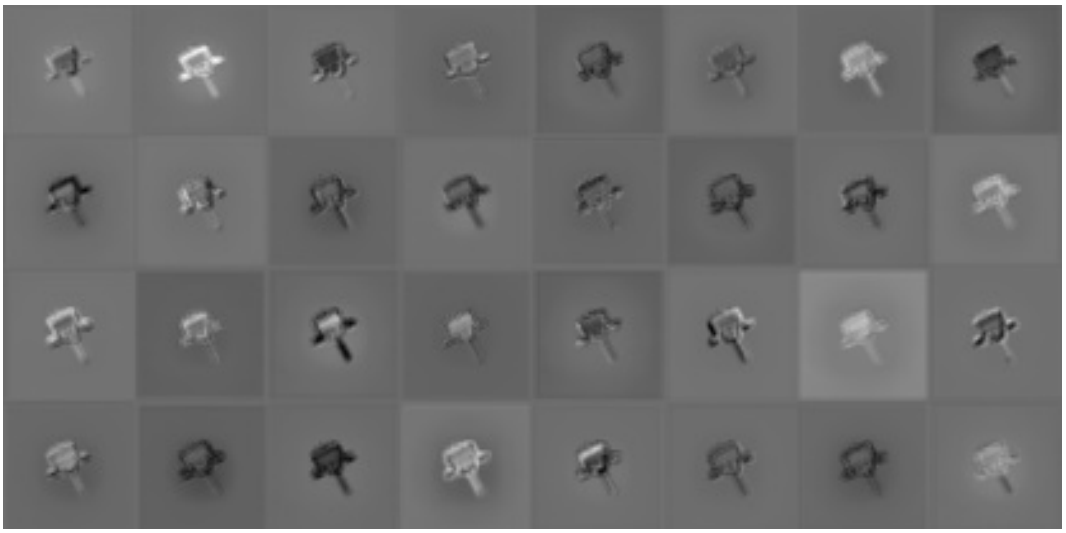


***Figure 6.26*. Visualisations of the conv_3 layer of *classNet_V4_16*, given an input sample of object kT001.**

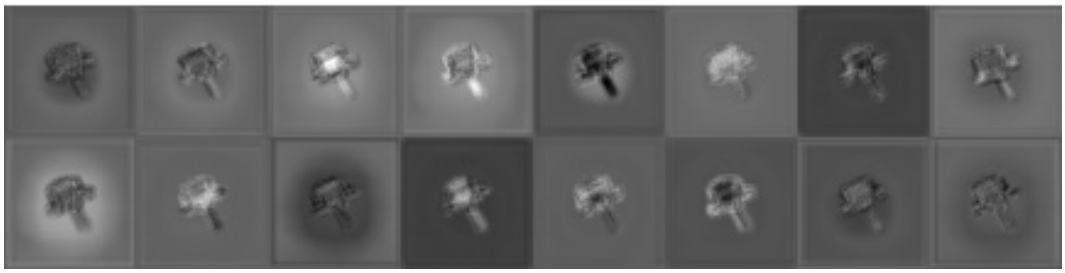


***Figure 6.27*. Visualisations of the conv_4 layer of *classNet_V4_16*, given an input sample of object kT001.**

Chapter 7

Stage 2: Grasp detection

7.1 Generating training data

In this research, grasp detection is framed as a binary classification task, i.e. one positive class and one negative class. Each class is defined subjectively by a human operator, with the intention of reflecting grasp areas that facilitate successful or unsuccessful grasps when physically attempted. A greyscale variant of the STG1_v4_HDC_clip dataset was used to gather samples for the stage 2 grasp detection dataset. Table 6.2, Section 6.1 provides more information regarding this dataset. 790 × 790 bounding box samples I_{BB} within this set are rotated via a Sobel filter to obtain $I_{BBR_{1\,..\,4}}$. See Section 4.3 for more information related to this process. The classification task employed in this stage of the proposed methodology is rotation-variant. Sun, Yu, Liu and Gu demonstrated better performance by rotating an input sample prior to classification, as opposed to training for rotation invariance [66]. As such, oriented samples are extracted manually from rotated space $I_{BBR_{1\,..\,4}}$ and saved to the relevant class folder. These samples relate to the top-down gripper placement window I_{RGW} as described in Section 4.2. Figure 7.1 and Figure 7.2 illustrate typical samples within the stage 2 dataset, corresponding to the positive class and the negative class, respectively.

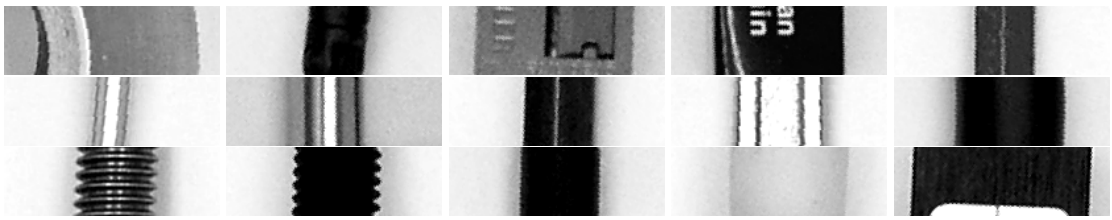


***Figure 7.1*. Examples of typical images associated with the *positiveGrasp* class within the stage 2 dataset. Object IDs from left to right, top to bottom: kH001, kH002, kH003, kH004, kH005, kT001, kT002, kT003, kT004, kT005, kC001, kC002, kC003, kC004, kC005.**

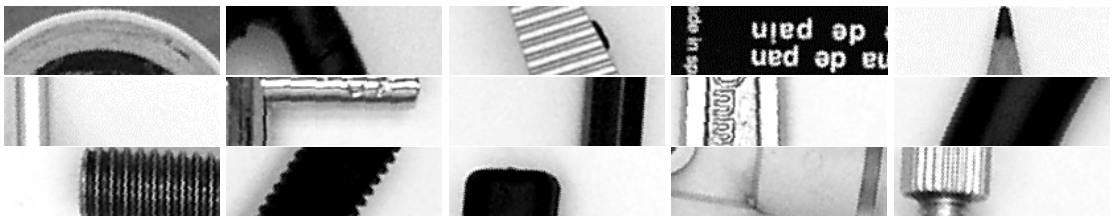


***Figure 7.2*. Examples of typical images associated with the *negativeGrasp* class within the stage 2 dataset. Object IDs from left to right, top to bottom: kH001, kH002, kH003, kH004, kH005, kT001, kT002, kT003, kT004, kT005, kC001, kC002, kC003, kC004, kC005.**

The stage 2 dataset is comprised of training samples for 2 classes, generated exclusively from the 15 objects related to the known object series, as documented by the STG1_v4_HDC_clip dataset. Note that the 16th class is not included, as no grasps can be associated with windows that do not contain an object. Stage 2 is responsible for the generation of potential grasps for further refinement. Samples within this set can be manually annotated, as noise introduced by the human operator is mitigated in the stage following. Additional redundancy was added to the stage 2 dataset in the form of bias.

Imposing an imbalance in sample size within a dataset skews classification toward the class with the most data. Therefore, the number of false positives produced by consequent networks can be reduced by increasing the number of negative samples. False negatives are of less importance and simply reduce the number of candidates generated for later assessment. To this end, the stage 2 dataset was biased toward the negative class—which has been shown to reduce the number of false positives when binary classification is employed [206].

250 parent images were randomly selected without replacement per class from the greyscale STG1_v4_HDC_clip dataset, for a total of 3,750 images. Each image was rotated to the top 4 maxima from a Sobel response, resulting in 1,000 rotated images per class, for a total of 15,000 rotated images. 1 positive and 2 negative samples were collected from each image. An example of this collection process is shown in Figure 7.3. Therefore, 15,000 positive samples and 30,000 negative samples are associated with each respective class.

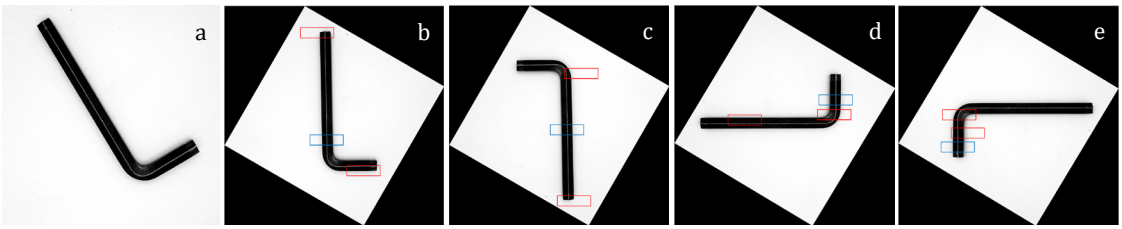


***Figure 7.3*. Example of parent image rotated by the top 4 maxima from a Sobel filter. (a)—input image selected from STG1_v4_HDC_clip. (b), (c), (d), (e)—rotated versions of input image. 1 positive grasp sample is annotated per rotated image, denoted in blue. 2 negative samples are annotated per rotated image, denoted in red.**

An additional 8,500 positive samples were taken from random images from the STG1_v4_HDC_clip dataset, without replacement. These images were rotated to one of the 4 maxima randomly, prior to manual annotation. 17,000 negative samples were also collected in this manner. In total, 23,500 raw samples are associated with the positive class and 47,000 raw samples are associated with the negative. Sample size was set to 164×52, according to grasping rectangle size I_{RGW}. These dimensions were found to correspond to the maximal opening of the implemented gripper, in pixels. Chapter 9.3 provides more information related to the acquisition of I_{RGW} size. Sample pixel intensity levels range from $[0\,..\,255]$. To double the amount of training data, all samples were rotated by 180°—bringing the total number of samples within the stage 2 dataset to 141,000. Figure 7.4 illustrates the generation of additional samples via 180°rotation.

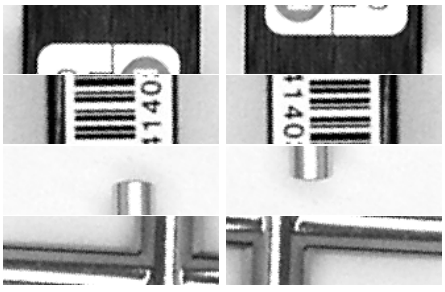


***Figure 7.4*. Illustration of additional training samples generated from an input sample. The left column contains the original sample. The right column contains the generated samples, rotated by 180°.**

The stage 2 dataset register is tabulated in Table 7.1. Earlier datasets are reproduced in Table 7.2, Table 7.3 and Table 7.4.

***Table 7.1*. Stage 2 dataset register (STG2_v4). Samples are greyscale. Sample size is set to 164 × 52.**

Class designation	object ID	Description	Number samples
positiveGrasp	kH001	Sellotape adhesive tape	2,000
	kH002	Figurine	2,000
	kH003	Pencil sharpener	2,000
	kH004	Factis eraser	2,000
	kH005	Pencil	2,000
	kT001	Electrical switch	2,000
	kT002	Small cross wrench	2,000
	kT003	Large hex key	2,000
	kT004	Small adjustable wrench	2,000
	kT005	Small side cutters	2,000
	kC001	Small silver bolt	2,000
	kC002	Rod end bearing	2,000
	kC003	Pull start handle	2,000
	kC004	Pneumatic T-splitter	2,000
	kC005	XCPC pneumatic valve	2,000
	-	Random selections from known object series	17,000
			47,000
negativeGrasp	kH001	Sellotape adhesive tape	4,000
	kH002	Figurine	4,000
	kH003	Pencil sharpener	4,000
	kH004	Factis eraser	4,000
	kH005	Pencil	4,000
	kT001	Electrical switch	4,000
	kT002	Small cross wrench	4,000
	kT003	Large hex key	4,000
	kT004	Small adjustable wrench	4,000
	kT005	Small side cutters	4,000
	kC001	Small silver bolt	4,000
	kC002	Rod end bearing	4,000
	kC003	Pull start handle	4,000
	kC004	Pneumatic T-splitter	4,000
	kC005	XCPC pneumatic valve	4,000
	-	Random selections from known object series	34,000
			94,000
			141,000

***Table 7.2*. Early stage 2 dataset register (STG2_v3). Samples are greyscale. Sample size is set to 164 × 52.**

Class designation	object ID	Description	Number samples
positiveGrasp	kH005	Pencil	1,000
	kT001	Electrical switch	1,000
	kT003	Large hex key	1,000
	kT004	Small adjustable wrench	1,000
	kT005	Small side cutters	1,000
	kC001	Small silver bolt	1,000
	kC002	Rod end bearing	1,000
	kC003	Pull start handle	1,000
	kC004	Pneumatic T-splitter	1,000
	kC005	XCPC pneumatic valve	1,000

	-	Random selections from known object series	1,000
			11,000
negativeGrasp	kH005	Pencil	2,000
	kT001	Electrical switch	2,000
	kT003	Large hex key	2,000
	kT004	Small adjustable wrench	2,000
	kT005	Small side cutters	2,000
	kC001	Small silver bolt	2,000
	kC002	Rod end bearing	2,000
	kC003	Pull start handle	2,000
	kC004	Pneumatic T-splitter	2,000
	kC005	XCPC pneumatic valve	2,000
	-	Random selections from known object series	2,000
			22,000
			33,000

***Table 7.3*. Early stage 2 dataset register (STG2_v2). Samples are greyscale. Sample size is set to 164 × 52.**

Class designation	object ID	Description	Number samples
positiveGrasp	kH005	Pencil	1,000
	kT001	Electrical switch	1,000
	kT003	Large hex key	1,000
	kT004	Small adjustable wrench	1,000
	kT005	Small side cutters	1,000
	kC001	Small silver bolt	1,000
	kC002	Rod end bearing	1,000
	kC003	Pull start handle	1,000
	kC004	Pneumatic T-splitter	1,000
	kC005	XCPC pneumatic valve	1,000
			10,000
negativeGrasp	kH005	Pencil	1,000
	kT001	Electrical switch	1,000
	kT003	Large hex key	1,000
	kT004	Small adjustable wrench	1,000
	kT005	Small side cutters	1,000
	kC001	Small silver bolt	1,000
	kC002	Rod end bearing	1,000
	kC003	Pull start handle	1,000
	kC004	Pneumatic T-splitter	1,000
	kC005	XCPC pneumatic valve	1,000
			10,000
			20,000

***Table 7.4*. Legacy stage 2 dataset register (STG2_v1). Samples are greyscale. Sample size is set to 280 × 150.**

Class designation	object ID	Description	Number samples
positiveGrasp	kC001	Small silver bolt	300
	kT003	Large hex key	300
	-	Small screwdriver	300
			900
negativeGrasp	kC001	Small silver bolt	300

	kT003	Large hex key	300
	-	Small screwdriver	300
			900
			1,800

7.2 Learning to detect grasps

Learning to grasp was approached heuristically, as per the procedures used to deduce network and dataset properties for classification in Chapter 6.2. 52 tentative networks were trained with variations in learning rate, batch size, model architecture, dataset construction, data dimensionality, training iterations and momentum optimisers. Early on, the STG2_v2 dataset was established (Table 7.3, Section 7.1). This set consisted of 20,000, 164×52 greyscale samples, geared toward binary classification. 10,000 samples relate to the *positiveGrasp* class, which aims to portray graspable areas. 10,000 samples are associated with the *negativeGrasp* class. This class relates to areas that are not graspable or may result in grasp failures when implemented. All samples within this dataset were generated by a human operator. *graspNet_V2_10* was a notable model trained using the STG2_v2 dataset. 70% of the data in this set was used for training. The remaining 30% was used for validation. *graspNet_V2_10* was trained for 20 epochs with a mini-batch size of 50. The initial learning rate was set to 5×10^{-4}. *graspNet_V2_10* correctly classified 97.8% of the training set and 97.8% of the validation set. The architecture of this model is shown in Figure 7.5.

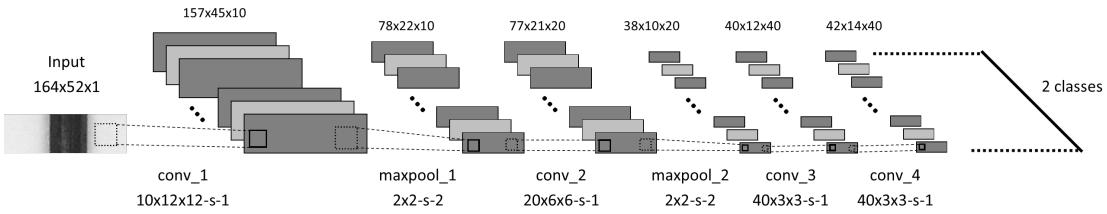


***Figure 7.5. graspNet_V2_10* architecture. The first convolutional layer filters the $164 \times 52 \times 1$ input image with 10 kernels of size 12×12 and a stride of 1 pixel. The second convolution takes as input the response-normalised and pooled output of the first convolutional layer, filtered with 20 kernels of size 6×6 for each of the resultant input layers. The third convolutional layer has 40 kernels of size 3×3, connected to the normalised and pooled previous convolution. The final convolutional layer also has 40 kernels of size 3×3. The fully connected layer has 2 neurons, one for each output class. In total, this network has 77,612 trainable parameters.**

During training it became evident that the STG2_v2 dataset tended to produce classifiers that suffered from type 1 errors, i.e. false positives. The confusion matrix for *graspNet_V2_10*, calculated from the validation set, is shown in Figure 7.6.

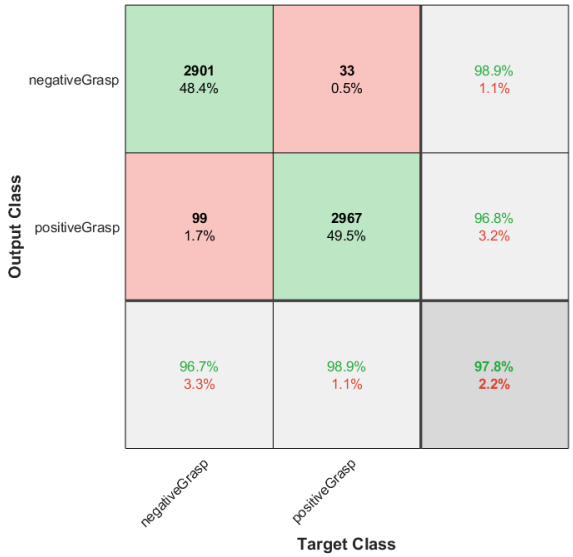


***Figure 7.6. graspNet_V2_10* confusion matrix, computed from the STG2_v2 dataset. Note that the axes for this confusion plot are inverted relative to normal convention due to the format applied by Matlab.**

The confusion matrix in Figure 7.6 illustrates that most errors made by the *graspNet_V2_10* network are false positives, i.e. samples that belong to the *negativeGrasp* class have been misclassified as belonging to the *positiveGrasp* class. In the context of the grasp generation task, misclassifying a positive sample as a *negativeGrasp* is inconsequential and simply reduces the number of candidates generated for the grasp selection phase of the proposed methodology. Conversely, false positives may be problematic and should be minimised. Ideally, no false positives should be present in the candidate grasp matrix, as this relies on the rejection of these samples by the stage 3 component prior to implementation. To address this issue and add further redundancy, the number of negative samples within the STG2_v2 dataset were doubled—resulting in the STG2_v3 dataset. This dataset contains 11,000 *positiveGrasp* samples and 22,000 *negativeGrasp* samples, for a total of 33,000 training samples. Table 7.2, Chapter 7.1 tabulates the STG2_v3 dataset. As per ML theory, manually disproportioning a dataset in this manner biases the resulting networks toward the desired class [135, 206]. For this specific task, network features should tend toward the *negativeGrasp* class, such that harsher criteria are applied to sample areas that belong to the positive class, i.e. trained networks are more particular when classifying a sample as *positiveGrasp*.

44 networks were trained using the STG2_v3 dataset. *graspNet_V3_10* denotes a notable network trained using 70% of the data in the STG2_v3 dataset. The remaining data was used for validation. The architecture for this network is identical to *graspNet_V2_10*. Moreover, this network was also trained for 20 epochs with a mini-batch size of 50 and initial learning rate of 5×10^{-4}. *graspNet_V3_10* correctly classified 98.01% of the training set and 98.22% of the validation set. To investigate the effect of dataset bias, this network was assessed using the STG2_v2 dataset. *graspNet_V3_10* scored 98.15% on the STG2_v2 training set and 98.4% on the STG2_v2 validation set. The confusion matrix for *graspNet_V3_10*, assessed using the STG2_v2 validation set, is shown in Figure 7.7.

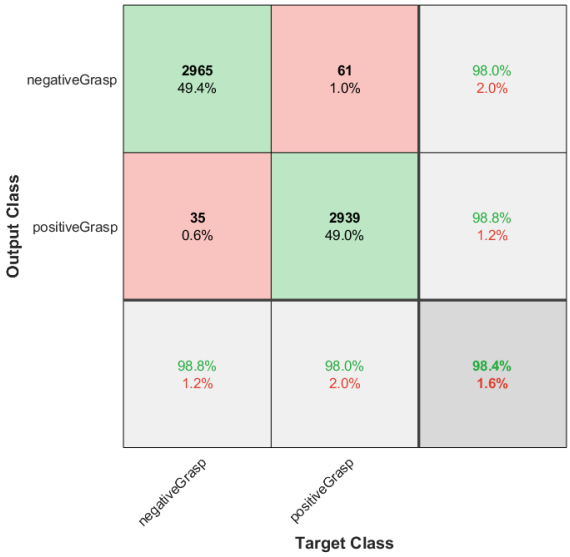


***Figure 7.7. graspNet_V3_10* confusion matrix, computed from the STG2_v2 dataset. Note that the axes for this confusion plot are inverted relative to normal convention due to the format applied by Matlab.**

An identical validation set comprised of 6,000 samples from STG2_v2 was used between networks to assess the effect of dataset bias. *graspNet_V3_10* showed better overall performance. The confusion matrix for this network (Figure 7.7) illustrates that *graspNet_V3_10* is biased toward the negative class. This bias resulted in a significant reduction in the number of misclassified negative samples, compared to *graspNet_V2_10*. Thus, demonstrating that the number of false positives produced by a network can be reduced by biasing its training set toward the negative class.

Over 200 networks were trained prior to settling on a dataset structure and network architecture suitable for the stage 2 grasp generation task. Many principles derived from the stage 1 classification task were employed while learning to detect grasps. The final dataset, STG2_v4, consisted of 141,000 samples, split into 1/3 (47,000) positive and 2/3 (94,000) negative samples, related to the *positiveGrasp* class and the *negativeGrasp* class, respectively. Data for this set was generated using the 15 objects within the known object series. The STG2_v4 dataset register can be found in Table 7.1, Chapter 7.1. The final network used to generate candidate grasps for the stage 2 task is denoted as *graspNet_V4_15*. Network architecture is illustrated in Figure 7.8.

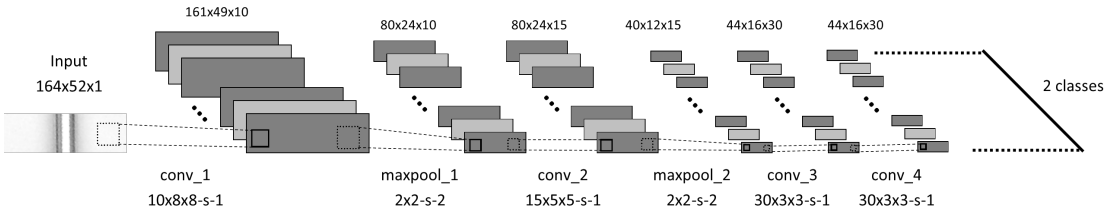


***Figure 7.8. graspNet_V4_15* architecture. The first convolutional layer filters the $164 \times 52 \times 1$ input image with 10 kernels of size 8×8 and a stride of 1 pixel. The second convolution takes as input the response-normalised and pooled output of the first convolutional layer, filtered with 15 kernels of size 5×5 for each of the resultant input layers. The third convolutional layer has 30 kernels of size 3×3, connected to the normalised and pooled previous convolution. The final convolutional layer also has 30 kernels of size 3×3. The fully connected layer has 2 neurons, one for each output class. In total, this network has 42,137 trainable parameters.**

graspNet_V4_15 was trained for 40 epochs, with a mini-batch size of 200, resulting in 22,560 iterations. The initial learning rate was set to 5×10^{-5}. Training and validation data were

taken from STG2_v4, split into 80% training data (112,800 images) and 20% validation data (28,200 images). It took approximately 85 minutes to train this network. *graspNet_V4_15* correctly classified 99.38% of the training set and 99.41% of the validation set—defined by the STG2_v4 dataset. Figure 7.9 shows the training and loss curves for this network. As noted in Section 6.2, the cited accuracies are merely estimates of real-world performance.

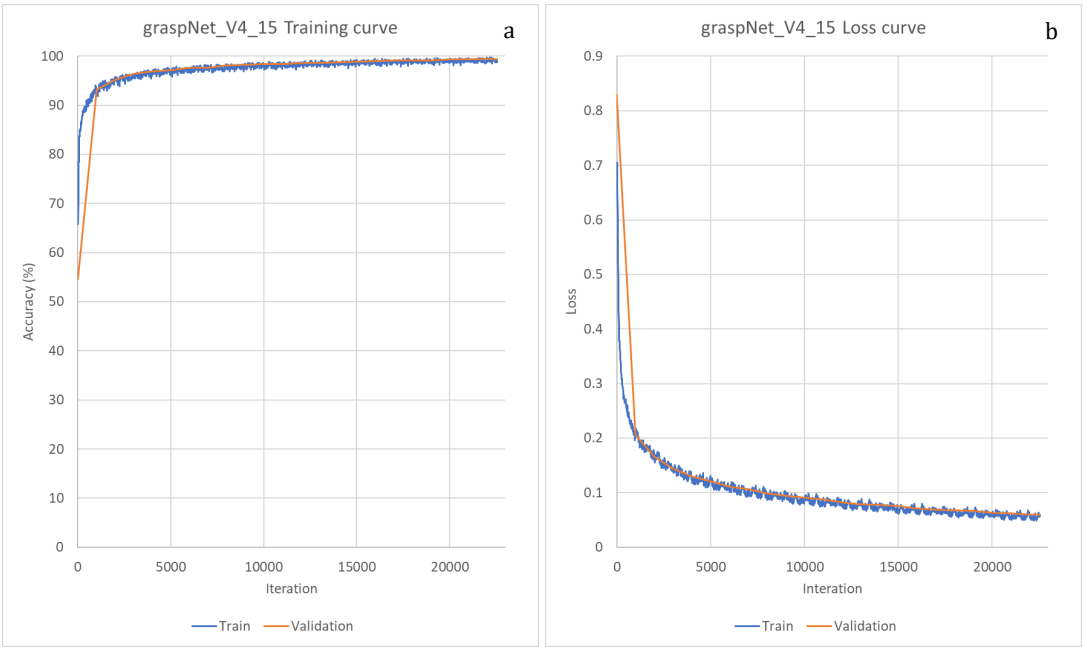


Figure 7.9. **(a)—training and validation curves of *graspNet_V4_15*, trained for 40 epochs with 80% of the STG2_v4 dataset. (b)—the corresponding loss curve of *graspNet_V4_15*.**

graspNet_V4_15 occupied 186 KB of disk space, with a total number of trainable parameters of 42,137. Time to classify for this network was 0.31 ms. Figure 7.10 illustrates several properties for this model in comparison to some of its notable predecessors.

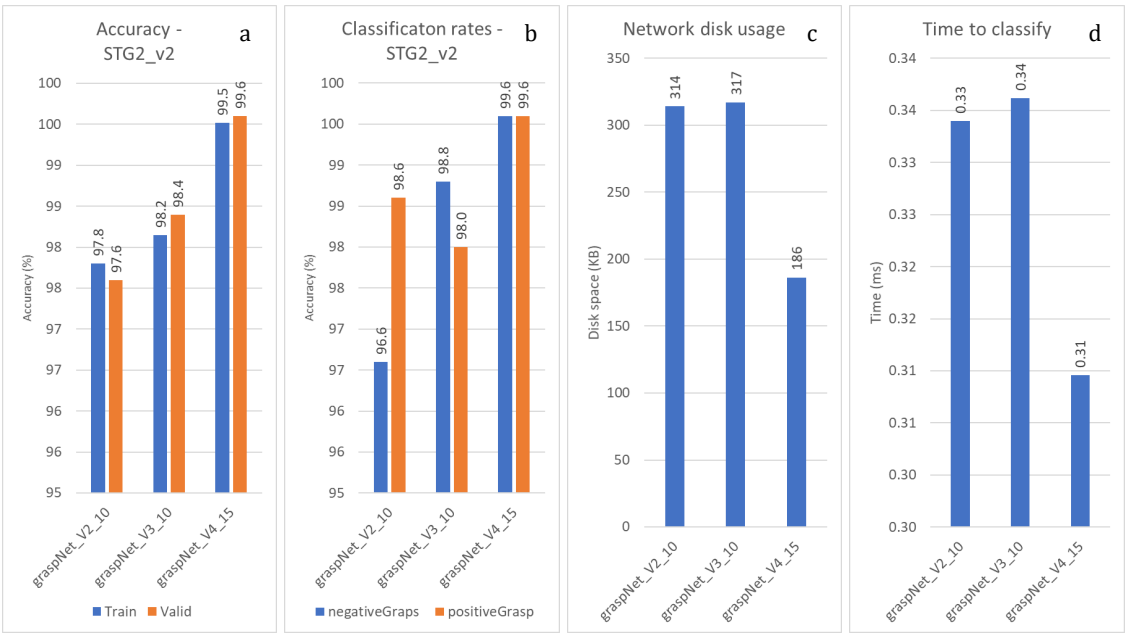


Figure 7.10. **(a)—STG2_v2 training and validation set accuracy by network. (b)—classification rates of various networks, calculated from STG2_v2. (c)— disk space occupied by various networks. (d)—time taken to classify a single input sample by network.**

The 6,000-sample validation set from STG2_v2 was used to assess the relative performance of *graspNet_V4_15* and the associated effect of dataset bias. *graspNet_V4_15* correctly classified 99.52% of the STG2_v2 training set and 99.60% of the validation set. The resulting confusion matrix is shown in Figure 7.11. The confusion matrix of *graspNet_V4_15* calculated from the STG2_v4 dataset is shown in Figure 7.12.

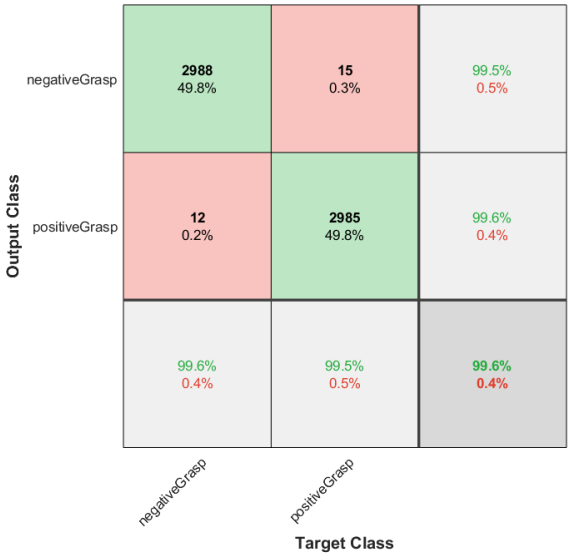


***Figure 7.11. graspNet_V4_15* confusion matrix, computed from the STG2_v2 dataset. Note that the axes for this confusion plot are inverted relative to normal convention due to the format applied by Matlab.**

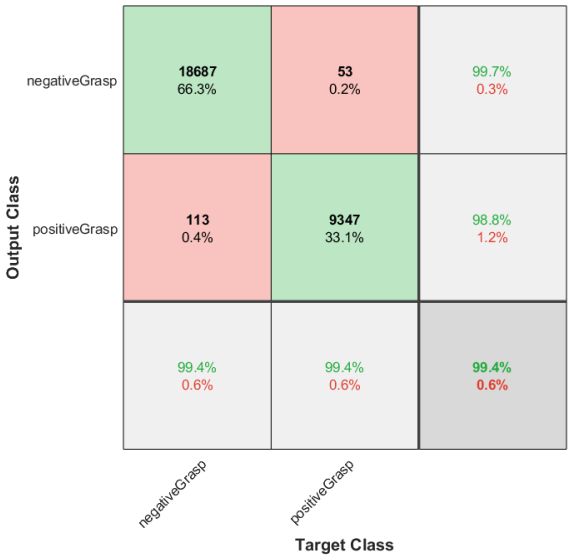


***Figure 7.12. graspNet_V4_15* confusion matrix, computed from the STG2_v4 dataset. Note that the axes for this confusion plot are inverted relative to normal convention due to the format applied by Matlab.**

Compared to *graspNet_V2_10* and *graspNet_V3_10*, *graspNet_V4_15* misclassified approximately the same number of positive samples as negative throughout STG2_v2. Moreover, the majority of misclassified samples are not false positives, which is ideal for added redundancy. Similarly, misclassification rates were approximately even between the two classes when tested on the STG2_v4 dataset (Figure 7.12). Note that the number of negative samples within this validation set is double the number of positive. The stage 2 task is relatively simple. With binary classification, 50% accuracy can be achieved by

random selection. This initial baseline is illustrated in the training curves in Figure 7.9—a, with the trained network starting out with approximately 50% classification and validation accuracy. The difficulty of the task is also reflected in the number of trainable parameters. *classNet_V4_16* for instance, has a relative increase in trainable parameters of approximately 18-fold. The activations of *graspNet_V4_15*, given the input sample provided in Figure 7.13, are visualised in Figures 7.14-7.17.

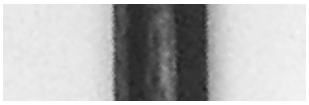


***Figure 7.13*. Sample generated from the pencilObject class (kH005). Softmax function output: 0.9989, classified as *positiveGrasp*.**

***Figure 7.14*. Visualisations of the conv_1 layer of *graspNet_V4_15*, given positive input sample of object kH005.**

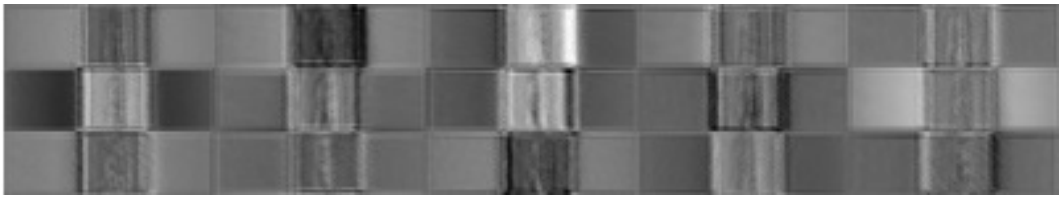


***Figure 7.15*. Visualisations of the conv_2 layer of *graspNet_V4_15*, given positive input sample of object kH005.**

***Figure 7.16*. Visualisations of the conv_3 layer of *graspNet_V4_15*, given positive input sample of object kH005.**

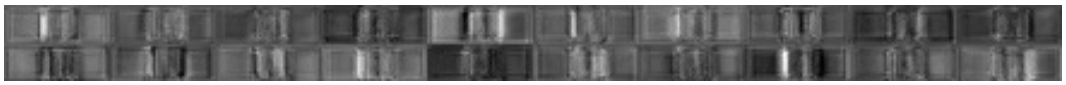


***Figure 7.17*. Visualisations of the conv_4 layer of *graspNet_V4_15*, given positive input sample of object kH005.**

Chapter 8

Stage 3: Grasp selection

8.1 Generating training data

Data for the stage 3, grasp selection component of the proposed methodology was collected through physical trials. Generating samples for this dataset was extremely slow. Objects within the known series placed haphazardly on the prototype conveyor were automatically located to the centre of the vision system, where numerous potential grasping areas were identified. A random sample from the candidate grasp matrix g was selected for implementation. Note that only samples with stage 2 softmax function scores $K_{I_{RGW}}$ of at least 0.7 were considered. The input scores S_c for the selected window are calculated and associated with the grasp. The object is translated into robot workspace, where it is grasped, lifted and placed back on the conveyor at its original position. The object is conveyed back to the vision system, where the resultant overlap score OS and orientation score OE may be calculated. Thus, input features and their consequent output were established—a typical structure for network training.

10 input features described by score matrix S_c were measured for each selected grasping window, prior to implementation. Chapter 4.2 provides more information regarding S_c. Outputs, OS and OE, were calculated post-grasp. Therefore, each attempted grasp sample within the stage 3 dataset can be formatted as vector:

$$K_{I_{RGW}}, S_{ss}, S_{cs}, S_{lsl}, S_{lsr}, S_{ptp}, S_{COG}, S_{svol}, S_{shs}, S_{sws}, OS, OE, pass/fail \quad \textbf{(8.1)}$$

where $pass/fail$ is a binary feature labelled by the user for each attempt. This metric relates to the traditional definition of grasp outcome. A grasp attempt was labelled as $pass$, or 1, if the object could be lifted vertically to a height of 15 cm and held stationary for at least 10 seconds without falling. The object had to be solely supported by the gripper, so that no other part of the object touched any other hardware. All other outcomes were labelled as $fail$, or 0.

Initially, 500 samples were collected with an earlier version of the methodology that utilised the *graspNet_V3_10* network to generate candidate grasps. Note that the prototype hardware had been finalised at this stage. 50 trials were conducted for each of the 10 objects within the known series, for a total of 500 samples. The STG3_v1 dataset register is available in Table 8.1.

***Table 8.1*. Early stage 3 dataset register (STG3_v1).**

object ID	Description	Number samples
kH005	Pencil	50
kT001	Electrical switch	50
kT003	Large hex key	50
kT004	Small adjustable wrench	50
kT005	Small side cutters	50

kC001	Small silver bolt	50
kC002	Rod end bearing	50
kC003	Pull start handle	50
kC004	Pneumatic T-splitter	50
kC005	XCPC pneumatic valve	50
		500

Selection stage networks trained with the STG3_v1 dataset were used in an initial round of trials to assess the performance of the 3-stage methodology. For more information regarding experimentation, refer to Chapter 9. To assess the proposed methodology more robustly, new data was collected utilising all 15 objects within the known series. 100 grasp trials were attempted for each object, resulting in 1,500 samples. Candidate grasps were generated by the *graspNet_V4_15* network. Due to the time-consuming nature of data collection, the STG3_v1 dataset and 1,500 samples were amalgamated to form the extended dataset STG3_v3. STG3_v3 consisted of 2,000 samples, each containing input score vector S_c and resultant output features OS, OE and $pass/fail$. The STG3_v3 dataset register is shown in Table 8.2.

***Table 8.2*. Stage 3 dataset register (STG3_v3).**

object ID	Description	Number samples
kH001	Sellotape adhesive tape	100
kH002	Figurine	100
kH003	Pencil sharpener	100
kH004	Factis eraser	100
kH005	Pencil	100
kT001	Electrical switch	100
kT002	Small cross wrench	100
kT003	Large hex key	100
kT004	Small adjustable wrench	100
kT005	Small side cutters	100
kC001	Small silver bolt	100
kC002	Rod end bearing	100
kC003	Pull start handle	100
kC004	Pneumatic T-splitter	100
kC005	XCPC pneumatic valve	100
		1,500
-	Additional samples from STG3_v1	500
		2,000

Input score averages from S_c by dataset are illustrated in Figure 8.1. The average output scores OS, OE and $pass/fail$ by dataset, are illustrated in Figure 8.2.

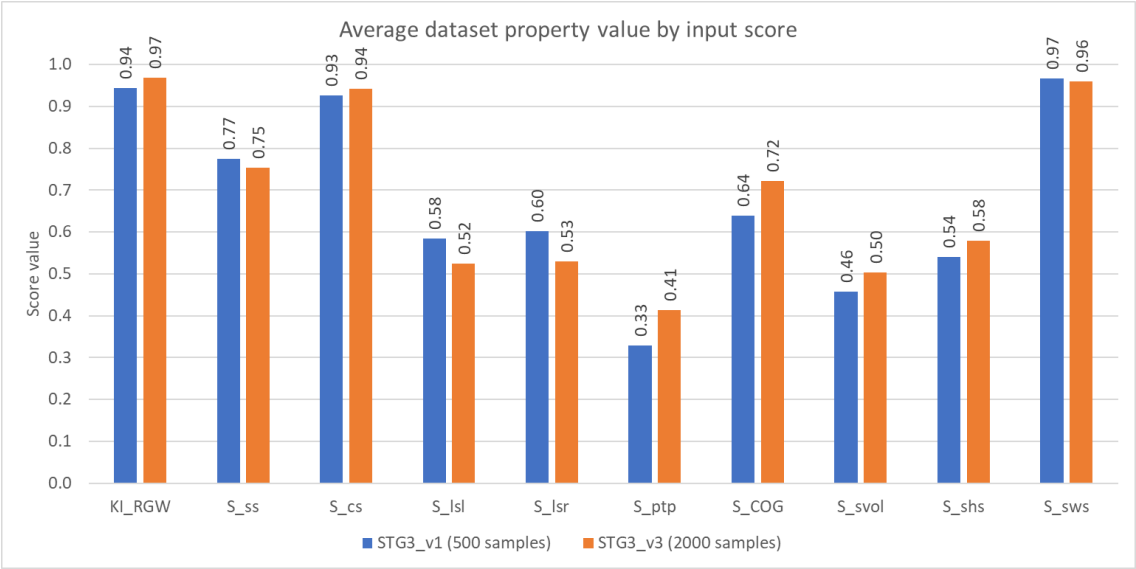


***Figure 8.1*. Average value of various input scores S_c by dataset.**

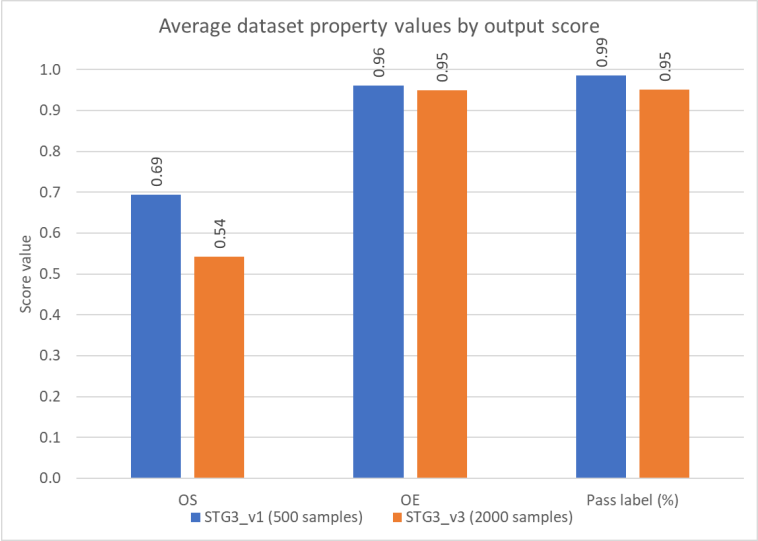


***Figure 8.2*. Average value of various output scores by dataset.**

The output scores shown in Figure 8.2 suggest a potential run-time performance if the system were to randomly select grasps generated by the stage 2 component with $K_{I_{RGW}} \geq 0.7$. Many works throughout literature simply implement and trial a system at this point, though generally, highest $K_{I_{RGW}}$ or some other selection metric is employed [13, 58, 66, 70, 71, 73]. In this thesis however, additional networks are trained using this data to further optimise for the desired outcome, OS and OE.

Compared to the stage 1 and stage 2 classification tasks mentioned in Chapter 6 and Chapter 7, respectively, learning to predict continuous OS and OE scores from input score matrix S_c was difficult—due to noisy data and lack of clear relationships between input and output variables. Several notable data clusters relating input and output features are graphed in Figures 8.3, 8.4 and 8.5. Although the $pass/fail$ label distinctions between samples have been graphed, there may be no relationship between OS and $pass/fail$ or OE and $pass/fail$. Moreover, this relationship is not relevant when selecting samples to maximise for OS, OE outcomes.

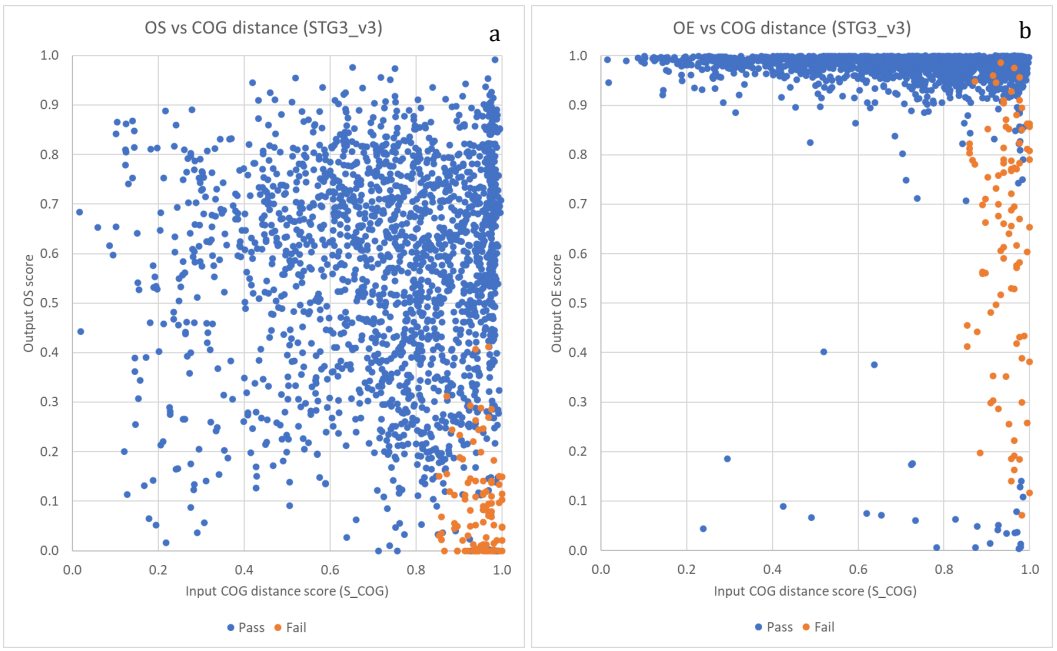


Figure 8.3. **(a)—scatter plot showing the measured output score OS from input score S_{COG}. (b)— scatter plot showing the measured output score OE from input score S_{COG}. Blue and orange points relate to samples labelled as $pass$ or $fail$, respectively. Data retrieved from the STG3_v3 dataset.**

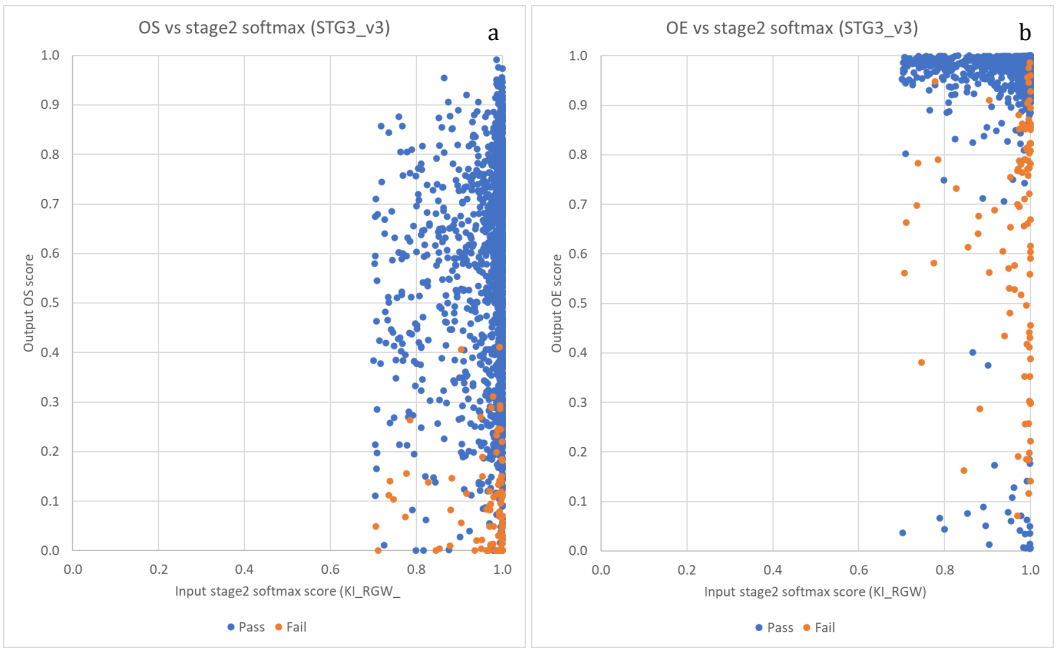


Figure 8.4. **(a)—scatter plot showing the measured output score OS from input score $K_{I_{RGW}}$. (b)— scatter plot showing the measured output score OE from input score $K_{I_{RGW}}$. Blue and orange points relate to samples labelled as $pass$ or $fail$, respectively. Data retrieved from the STG3_v3 dataset. Note that only samples with stage 2 softmax scores of higher than 0.7 were selected for implementation.**

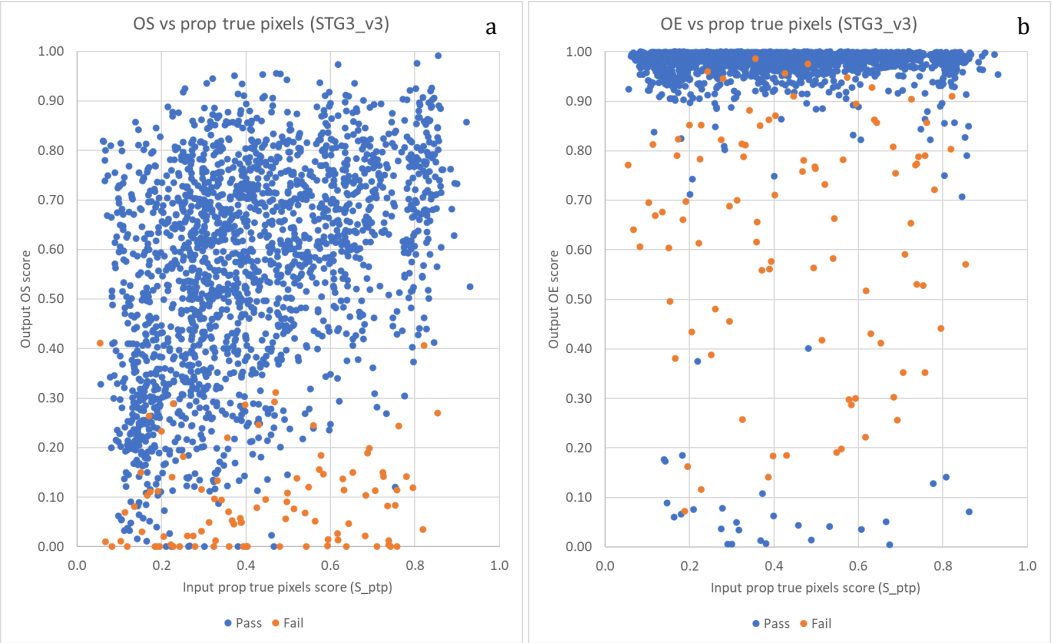


***Figure 8.5*. (a)—scatter plot showing the measured output score OS from input score S_{ptp}. (b)— scatter plot showing the measured output score OE from input score S_{ptp}. Blue and orange points relate to samples labelled as $pass$ or $fail$, respectively. Data retrieved from the STG3_v3 dataset.**

8.2 Learning to grasp

The stage 3 component of the proposed methodology selects a final grasp for implementation from the pool of candidate grasps generated by stage 2. To help facilitate selection, this learner was initially cast as a classifier through threshold intervals. Morales, Chinellato, Fagg and del Pobil discretised their continuous data through the definition of 5 classes: A, B, C, D and E [83], where A represents a grasped object that did not fall and was returned successfully to its initial position. E indicates that the system was not able to lift the object. They trialled 844 grasps, recording input and output features in terms of their stability metrics. Each sample was labelled with a reliability class based on the outcome. Consequently, they trained a KNN classifier to predict the reliability class (A, B, C, D, E) of grasp samples prior to implementation [83]. Thus, candidate grasps with a lower predicted probability of success could be avoided. Similarly, continuous output features OS and OE from the STG3_v3 dataset for each sample were grouped according to combined score OV, calculated as:

$$OV = \frac{OS + OE}{2} \tag{8.2}$$

The STG3_v3 dataset register is provided in Table 8.2, Section 8.1. 5 grade classes were established by the intervals shown in Table 8.3. Grade 5 represents an attempted grasp sample that produced poor OS and OE scores. Conversely, grade 1 indicates a grasp sample that resulted in high output scores.

***Table 8.3*. Intervals used to define 5 discrete classes for continuous variable *OV*.**

$Grade_5$	Interval
1	$0.8 \leq OV \leq 1.0$
2	$0.6 \leq OV < 0.8$
3	$0.4 \leq OV < 0.6$
4	$0.2 \leq OV < 0.4$
5	$0.0 \leq OV < 0.2$

Thus, grasp samples could be formatted as graded vector:

$$K_{I_{RGW}}, S_{ss}, S_{cs}, S_{lsl}, S_{lsr}, S_{ptp}, S_{COG}, S_{svol}, S_{shs}, S_{sws}, Grade_5 \tag{8.3}$$

where $Grade_5$ is the output feature related to inputs $K_{I_{RGW}}, S_{ss}, S_{cs}, S_{lsl}, S_{lsr}, S_{ptp}, S_{COG}, S_{svol}, S_{shs}$ and S_{sws}. 72 various classifiers were trained to predict $Grade_5$, with the top performing network achieving a mere 57.6% classification accuracy. This network is denoted as *gradeNet_V3_5*. *gradeNet_V3_5* was an SVM model, with a Gaussian kernel scale of 3.2. One-vs-one multi-class training was employed. 5-fold cross-validation was used to partition the dataset during training to help protect against overfitting. The relatively low classification accuracy achieved by *gradeNet_V3_5* was not suitable for implementation. Note that randomly selecting between the 5 available classes would result in the correct classification of 20% of samples. To address this issue, the distribution of $Grade_5$ with respect to the STG3_v3 dataset was investigated. The graded sample distribution when partitioning output *OV* into 5 classes is tabulated in Table 8.4 and visually illustrated in Figure 8.3.

***Table 8.4*. Sample distribution among graded output class $Grade_5$ from the STG3_v3 dataset.**

$Grade_5$ label	1	2	3	4	5
Number of samples	879	853	191	60	17
% of STG3_v3	44.0%	42.7%	9.6%	3.0%	0.85%

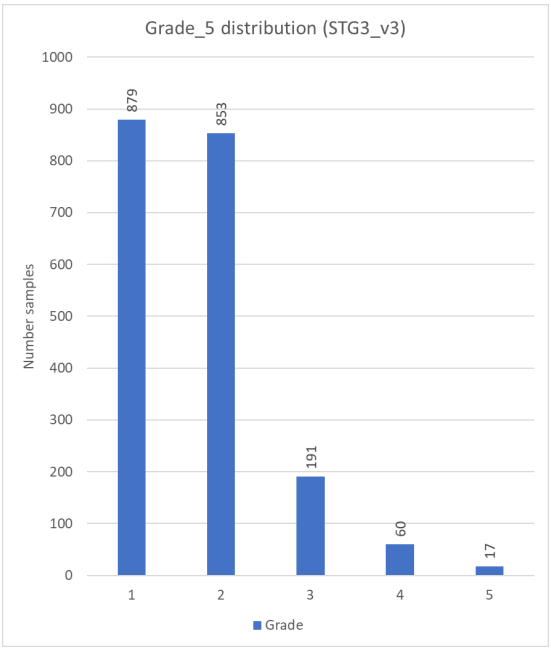


***Figure 8.6*. Visualisation of the sample distribution of graded output class $Grade_5$ from STG3_v3.**

This distribution indicates that assigning only 5 grades to combined score OV provides very little inter-class discrimination. Approximately 87% of all data within STG3_v3 fall within interval $0.6 \leq OV \leq 1.0$. Therefore, a graded approach with double the resolution was investigated. 10 output classes, $Grade_{10}$, were defined according to the intervals in Table 8.5.

***Table 8.5*. Intervals used to define 10 discrete classes for continuous variable OV.**

$Grade_{10}$	Interval
1	$0.9 \leq OV \leq 1.0$
2	$0.8 \leq OV < 0.9$
3	$0.7 \leq OV < 0.8$
4	$0.6 \leq OV < 0.7$
5	$0.5 \leq OV < 0.6$
6	$0.4 \leq OV < 0.5$
7	$0.3 \leq OV < 0.4$
8	$0.2 \leq OV < 0.3$
9	$0.1 \leq OV < 0.2$
10	$0.0 \leq OV < 0.1$

The resultant 10-class, graded vector associated with each sample was therefore formatted as:

$$K_{I_{RGW}}, S_{ss}, S_{cs}, S_{lsl}, S_{lsr}, S_{ptp}, S_{COG}, S_{svol}, S_{shs}, S_{sws}, Grade_{10} \quad \textbf{(8.4)}$$

24 various networks were trained to predict $Grade_{10}$ from dataset inputs $K_{I_{RGW}}, S_{ss}, S_{cs}, S_{lsl}, S_{lsr}, S_{ptp}, S_{COG}, S_{svol}, S_{shs}, S_{sws}$. A notable network trained for 10-way classification is denoted as *gradeNet_V3_10*. This network was based on the SVM model, with a Gaussian kernel. 5-fold cross-validation was used during training. This network performed worse than the *gradeNet_V3_5* model, scoring a classification accuracy of only 39.9%, as expected, since the classification difficulty is increased with double the number of classes. The graded sample distribution when partitioning output OV into 10 classes is tabulated in Table 8.6 and visually illustrated in Figure 8.4.

***Table 8.6*. Sample distribution among graded output class $Grade_{10}$ from the STG3_v3 dataset.**

$Grade_{10}$ label	1	2	3	4	5	6	7	8	9	10
Number of samples	189	690	513	340	142	49	30	30	11	6
% of STG3_v3	9.5%	34.5%	25.7%	17.0%	7.1%	2.5%	1.5%	1.5%	0.6%	0.3%

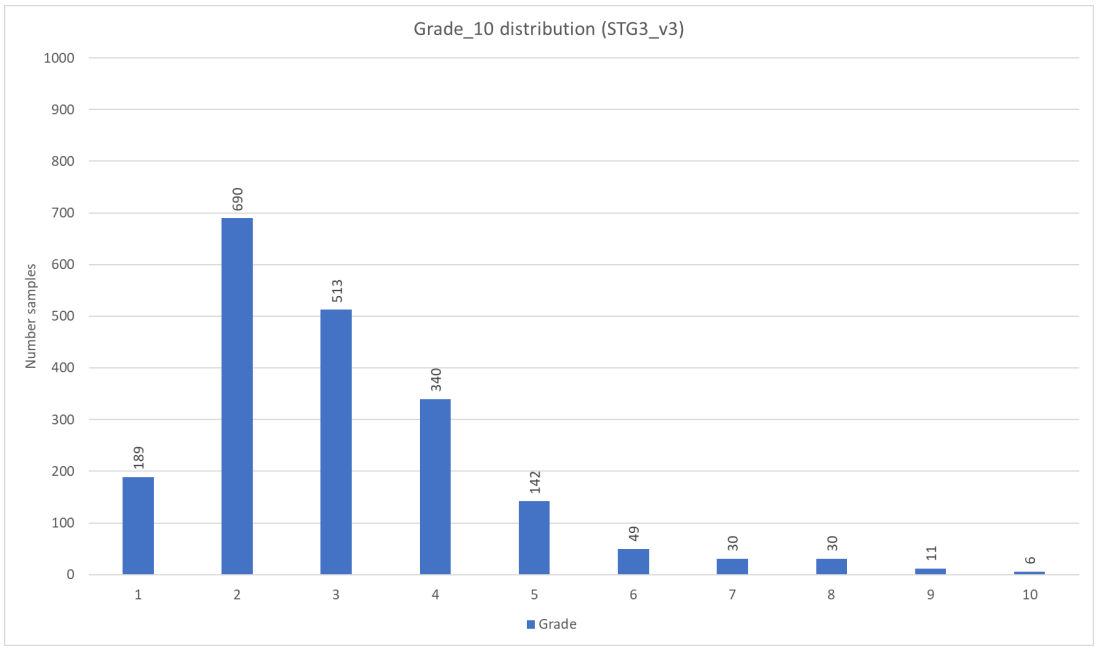


***Figure 8.7*. Visualisation of the sample distribution of graded output class $Grade_{10}$ from STG3_v3.**

Partitioning OV into 10 classes provided better inter-class discrimination than 5-way classification, although 34.5% of the data from STG3_v3 fell within a single classification: grade 2. Upon investigating the distribution of output scores OS and OE and combined score OV, it became evident that candidate grading by classification was an inappropriate approach toward grasp selection. The distributions of output scores OS, OE and OV from the STG3_v3 dataset are shown in Figure 8.8, Figure 8.9 and Figure 8.10, respectively.

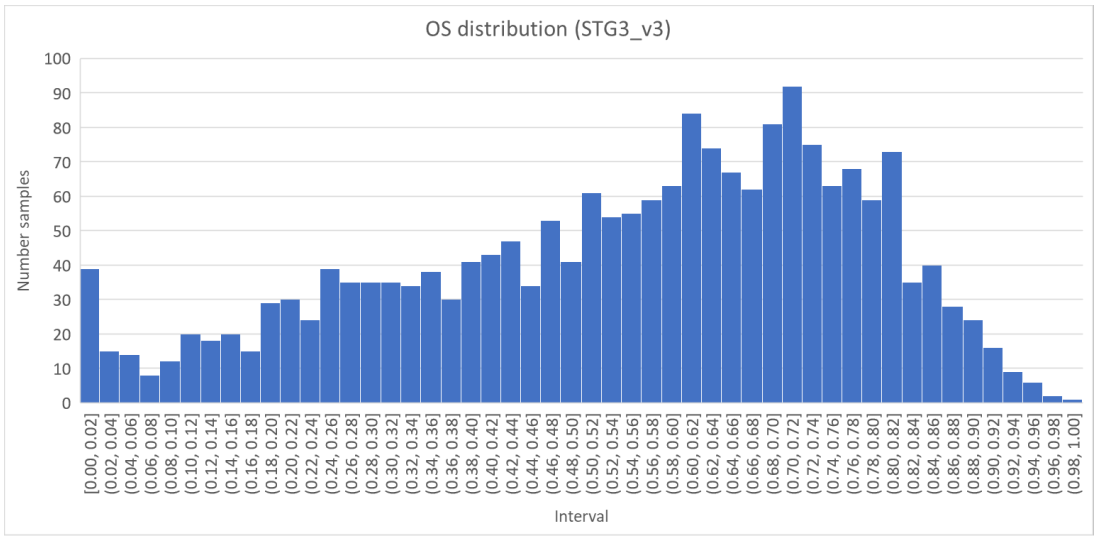


***Figure 8.8*. Visualisation of the sample distribution of output score OS from STG3_v3.**

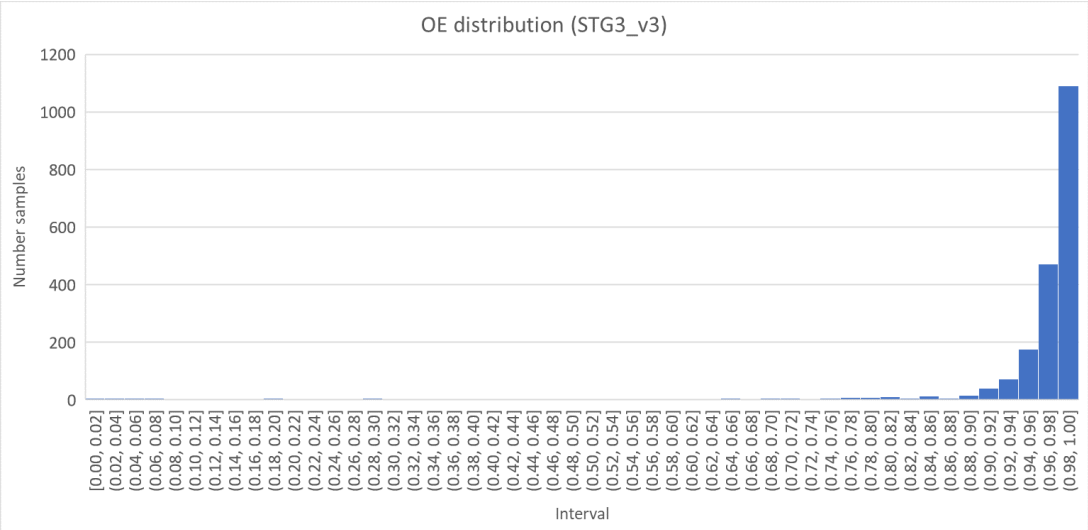


***Figure 8.9*. Visualisation of the sample distribution of output score OE from STG3_v3.**

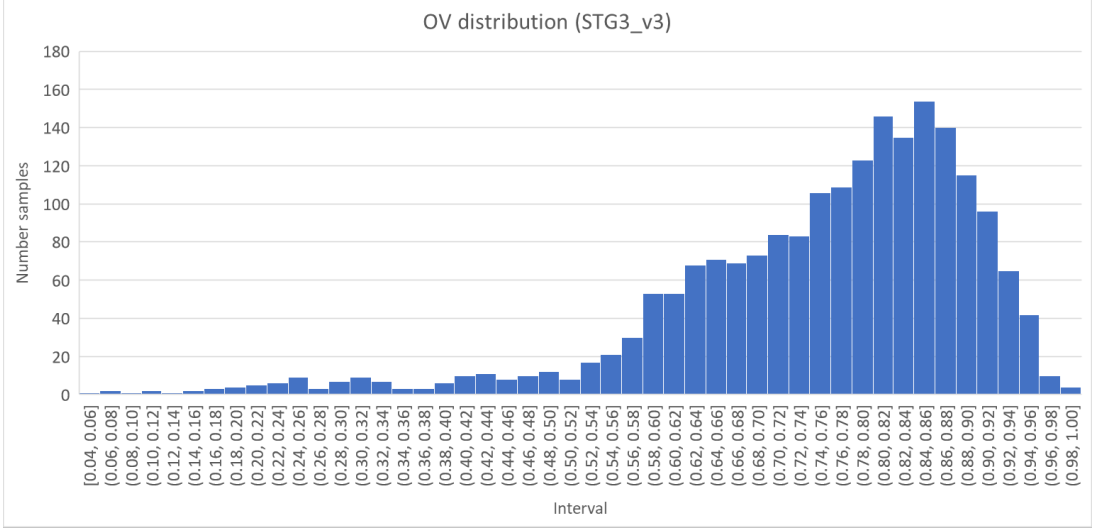


***Figure 8.10*. Visualisation of the sample distribution of combined output score OV from STG3_v3.**

The class distribution of $Grade_5$ (Figure 8.6) and $Grade_{10}$ (Figure 8.7) reflect a low-resolution representation of the distribution of OV illustrated in Figure 8.10. The grasp success rate ($pass/fail$) for samples recorded in the STG3_v3 dataset was approximately 95% (Figure 8.2, Section 8.1). This low rate of failure did not help to diversify the STG3_v3 dataset, resulting in significant skew toward higher OS and OE scores, shown in Figure 8.8 and Figure 8.9, respectively. To promote learned features that discriminate between class labels, a classifier should ideally have access to a dataset in which the number of samples between classes is even. To illustrate, suppose that a binary classifier was given the task of predicting a $pass/fail$ label for samples from the STG3_v3 dataset. Such a classifier could achieve 95% accuracy by simply labelling all samples as $pass$. To this end, alternatives to classification were investigated.

In classification schemes, new instances of data are categorised into discrete groups based on learned features about the associated inputs. In contrast, regression approaches aim to predict continuous outputs from learned relationships between predictors and their response. For a general overview of classification and regression analysis, the reader is

directed to the practical ML textbook by Raschka and Vahid [206]. In considering regression analysis, a correlation matrix was computed between each variable recorded in STG3_v3 to assess any potential linear relationships between features. This matrix is graphically illustrated in Figure 8.11. Figure 8.12 highlights the relationships between predictors and response variables.

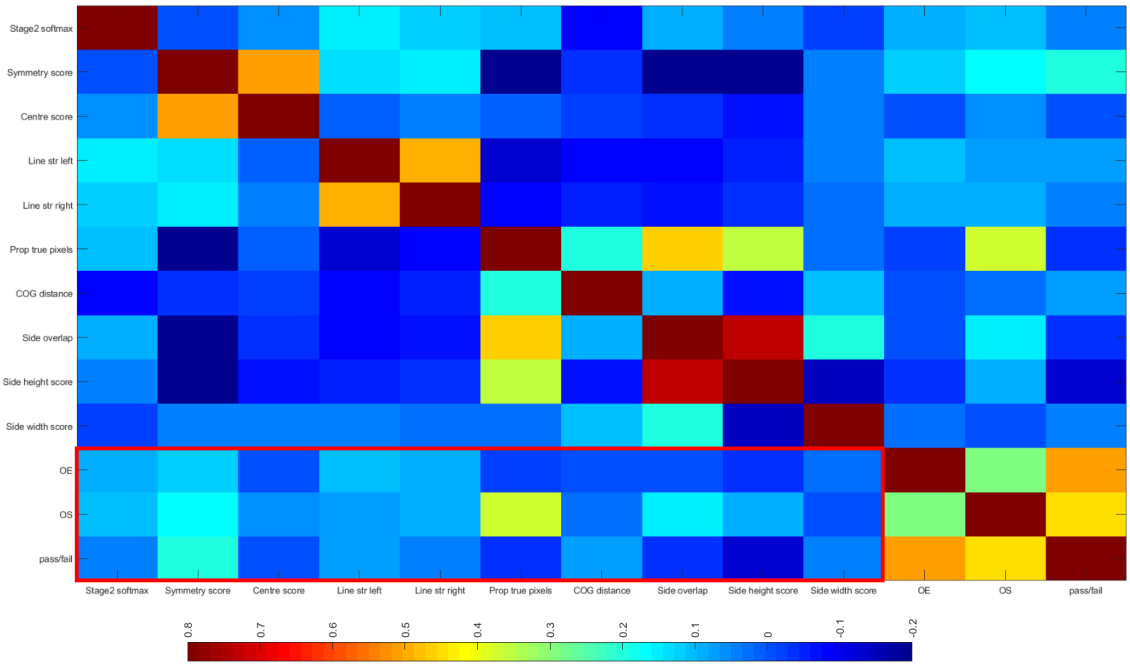


***Figure 8.11*. Linear correlation matrix between all variables in the STG3_v3 dataset. Relationships between input and output features are framed by the red bounding box. The scale for this matrix is [-0.2, 0.8].**

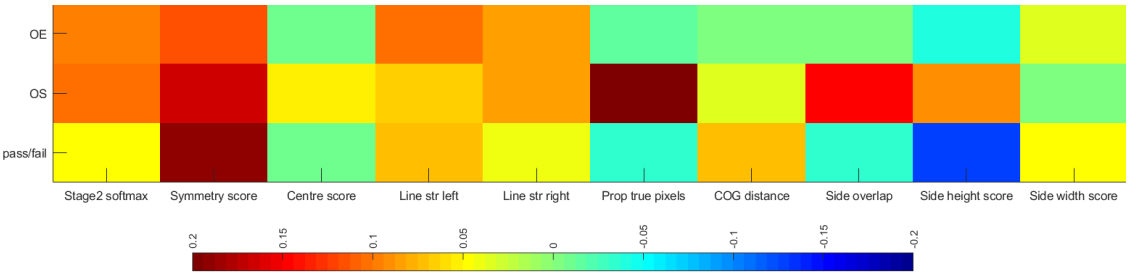


***Figure 8.12*. Snippet of correlation matrix between input and output features. The scale for this matrix is [-0.2, 0.2].**

No particularly strong linear correlations were present between variables in the STG3_v3 dataset. Some notable relationships between input features include line str left/line str right (S_{lsl}/S_{lsr}), symmetry score/centre score (S_{ss}/S_{cs}) and side overlap/side height score (S_{svol}/S_{hs}). The correlation coefficients between S_{lsl}/S_{lsr}, S_{ss}/S_{cs} and S_{svol}/S_{hs} were 0.49, 0.52 and 0.74, respectively (Figure 8.11). The relationship between the symmetry of a grasp S_{ss} within window I_{RGW} and the centre score S_{cs} was expected. As the assessed area tends toward perfect symmetry, the mid-location of the grasp tends toward the centre of I_{RGW}. Thus, S_{ss} and S_{cs} are related and respond similarly to the same influence. Generally, the clarity of the grasping area within I_{RGW} was related to the assessed object. Darker objects tended to produce sharp edges, compared to lighter objects. As such, S_{lsl} and S_{lsr} respond in tandem to the resulting edge clarity of a specific object. Chapter 4.2 provides a comprehensive description related to the calculation procedure of the input scores recorded in the STG3_v3 dataset.

No single input variable responded linearly to outputs OS, OE or $pass/fail$, although a notable relationship existed between OS and the proportion of I_{RGW} filled with the graspable area S_{ptp}. The correlation coefficient between OS and S_{ptp} was 0.36. This may suggest a relationship between gripper contact area and the resultant output score, although the correlation is very weak. Notable relationships between output scores were $OS/pass/fail$ and $OE/pass/fail$. The relationship between output orientation score OE and grasp outcome $pass/fail$ was expected during trials because failed grasps tended to qualitatively introduce a significant amount of rotational error. The correlation coefficient between OE and $pass/fail$ was 0.51. Though no clear linear relationships between individual features were apparent within STG3_v3, framing the stage 3 component as a regression problem yielded improved results. Note that many non-linear regression learners exist. SVMs for instance, can be trained for classification or regression. For a comprehensive report related to support vector machines for classification and regression, refer to Gunn [389]. Moreover, multiple linear regression (MLR) algorithms fit multi-dimensional models that respond to the culmination of input variables.

Three separate regression models were trained using the STG3_v1 dataset to predict the resultant OS, OE and probability of $pass$ label, given input S_c. The networks are denoted as *OSpredNet_V1, OEpredNet_V1* and *passpredNet_V1*, respectively. Over 90 models were trained prior to these networks. All models were trained using 5-fold cross-validation. *OSpredNet_V1* achieved a root mean squared error (RMSE) of 0.17. This network utilised a regression tree with a leaf-size of 36. *OEpredNet_V1* and *passpredNet_V1* utilised MLR models, achieving RMSEs of 0.14 and 0.12, respectively. These initial networks were used in an early performance assessment trial of the proposed methodology. More information regarding details and the outcome of this trial are available in Chapter 10. Due to the time-consuming nature of data collection for the selection stage dataset, STG3_v1 consisted of only 500 samples. Consequently, building accurate models was difficult. To this end, the STG3_v3 dataset was collected.

In total, over 180 regression models were trained using STG3_v3 prior to settling on appropriate networks for selection. Three notable regression models were trained using 5-fold cross-validation and the 2,000 samples from STG3_v3. *OSpredNet_V3* achieved a RMSE of 0.07, *OEpredNet_V3* achieved a RMSE of 0.14 and *passpredNet_V3* achieved a RMSE of 0.08. *OSpredNet_V3* and *passpredNet_V3* utilised SVM models. *OEpredNet_V3* utilised MLR. *OSpredNet_V3* and *OEpredNet_V3* are used to predict the OS and OE outcome of hypothesis grasps, denoted as OS_{pred} and OE_{pred}, respectively. These scores are used in the proposed selection methodology, described in Section 4.3. The *passpredNet_V3* model is trained to predict the probability of a $pass$ label outcome, denoted as $pass_{pred}$, for a given sample based on the 10 input features from S_c. This network was trained for comparison purposes.

Chapter 9

Implementation and testing

9.1 Simulation

An experimental prototype was designed and constructed to evaluate and physically trial the proposed grasp synthesis methodology. The test-rig was modelled prior to construction, with consideration given to the vision system, enclosure size, lighting conditions, conveyor design, manipulator placement and robot workspace. The design is typical of a vision inspection station common in production line automation. Figure 9.1 illustrates the modelled test-rig. For an annotated version of this figure, refer to Figure 4.2—b, Chapter 4.1.

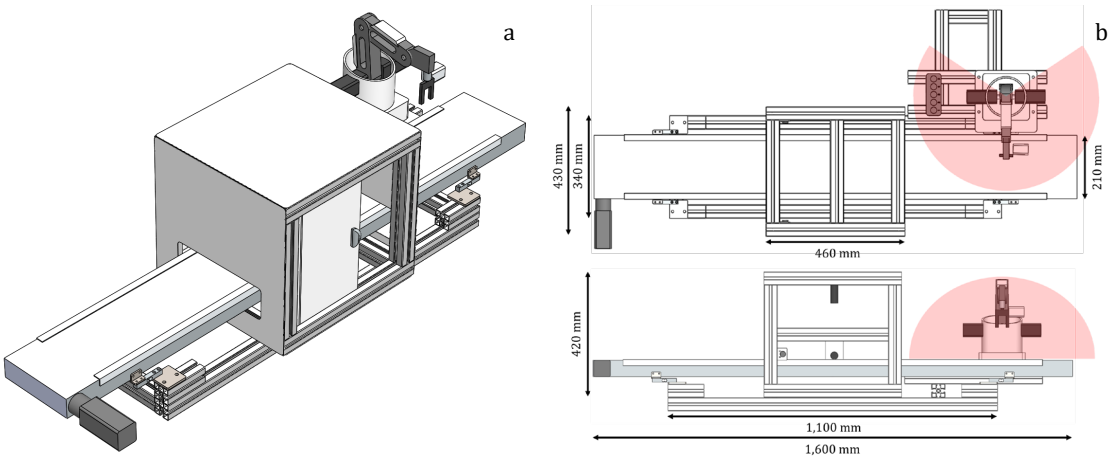


***Figure 9.1*. (a)—isometric view of the finalised test-rig model. (b)—top and side views of the model, with measurements and approximate robot workspace annotated.**

The enclosure was designed to measurements that optimised camera focal length and light dispersion characteristics, derived through simulation. The enclosure opening was constrained to help limit object size, congruent with robot payload, gripper size and gripper closure force. Moreover, simulations showed that interference from external light could be minimised by maximising outer sheet coverage. A lighting simulation was conducted in Solidworks Photoview 360. For a general review and description of this software, see the resource by Reyes [390]. For photorealistic rendering, lighting and cameras, the reader is referred to the comprehensive guide by Dassault Systèmes [391]. The lighting simulation is shown in Figure 9.2.

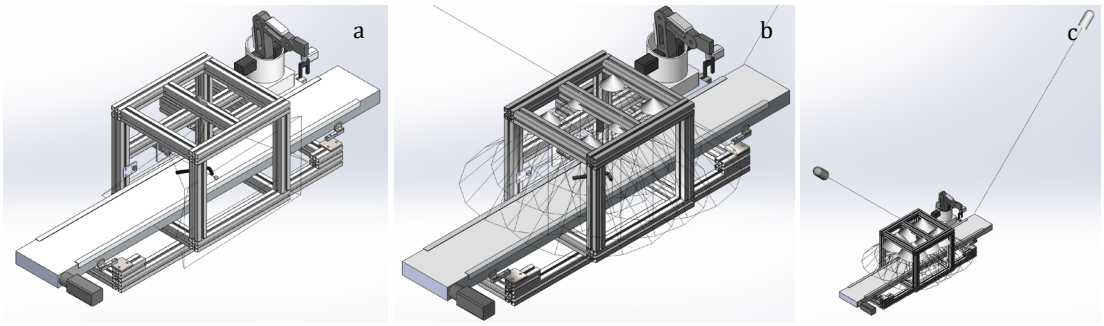


***Figure 9.2*. Lighting simulation utilising Solidworks Photoview 360. (a)—3D model used to simulate lighting conditions and camera placement. Top and side cameras annotated. (b)—6 diffuse LED lights annotated. (c)—6 diffuse LED lights and 2 ambient, directional lights annotated. Enclosure covers have been removed for clarity.**

6 diffuse LED light sources were added to the top of the enclosure to emulate LED strips with a plastic diffusor panel. Simulated lights were facing directly downward toward the conveyor, with brightness levels set to 0.3 $\frac{W}{m^2}$. Two directional light sources were added at 45° to each conveyor-enclosure entrance. Brightness levels were set to 1.0 $\frac{W}{m^2}$. Shadows and ambient occlusion were active. Material reflectivity properties were set according to Solidworks default values. No other ambient light was added. Shadow quality was set to 16. Intrinsic camera properties were defined according to their real values, documented by the manufacturer in Table 9.7, Section 9.2.2. Figure 9.3 shows an example of a simulation conducted with the finalised camera placement and lighting conditions. Simulated images are compared to their real counterpart, captured once the prototype had been constructed.

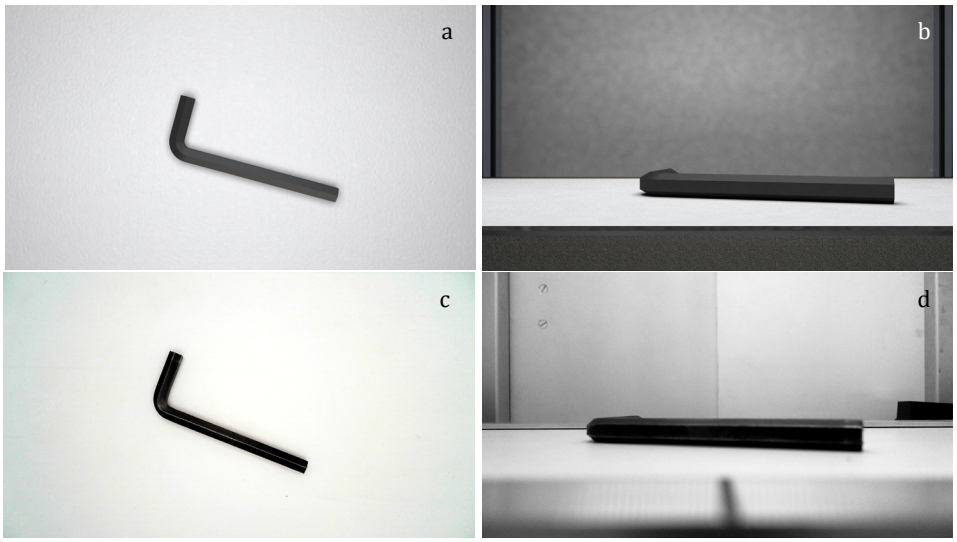


***Figure 9.3*. (a, b)—simulated images of the hex key object from top perspective (a) and side perspective (b). (c, d)—real images of the hex key object from top perspective (c) and side perspective (d).**

9.2 Hardware setup

The implemented prototype broadly consisted of a conveyor system, vision enclosure and robotic manipulator. The stepper motor driven conveyor rested on 4 load-cells, used to calculate the COG coordinates of an object, $x_{t_{COG}}$ and $y_{t_{COG}}$. A $460 \times 430 \times 420$ mm enclosure was constructed using 45×45 mm aluminium extrusion. The inside panels were matt white. 2 LED strips were mounted at the top of the enclosure, facing downward. A

plastic diffusor was added at the top of the enclosure, below the lighting. Two cameras were mounted to fulfil the methodology requirements discussed in Chapter 4. An educational robotic manipulator was mounted to the apparatus, with a workspace sufficient to grasp imaged objects resting on the conveyor. An integrated power and microcontroller system was employed to control the test-rig using a computer. A beam sensor was added inside the enclosure to provide a means of initial object detection so that an object could be translated to the centre of the vision system. Various images of the prototype are illustrated in Figure 9.4.

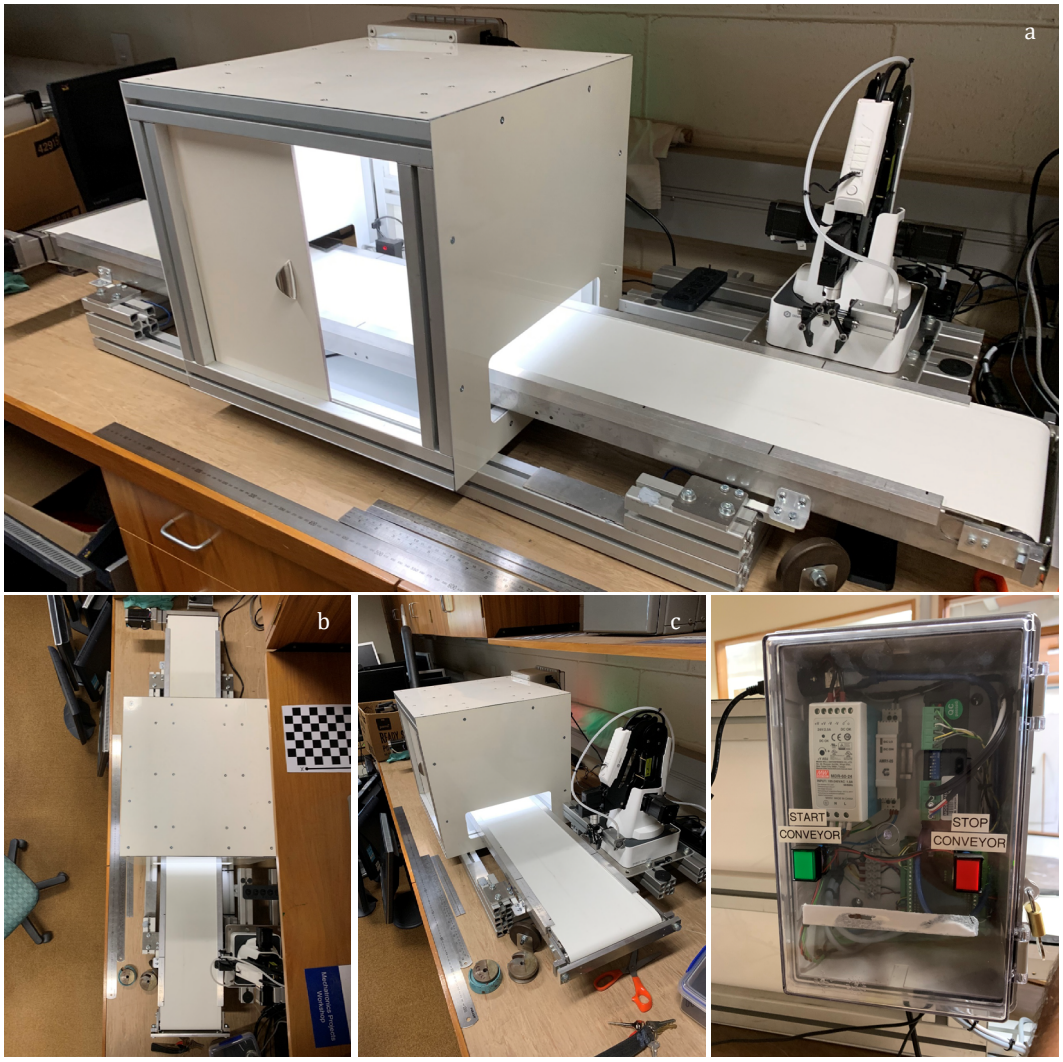


***Figure 9.4*. Various images of the test-rig. (a)—frontal view of the complete apparatus. (b)—top view of the apparatus. (c)—side view of the apparatus. (d)—control box containing the power system, load-cell amplifiers, stepper motor driver, LED driver and microcontroller.**

An NVIDIA Quadro K2200 was used to train and execute the ML-based components of the proposed methodology. Matlab was used for implementation. Table 9.1 and Table 9.2 provide more information related to computer hardware and software details, respectively.

***Table 9.1*. Computer hardware details.**

Component	Details
Graphics card	NVIDIA Quadro K2200 [392] • 4GB GDDR5

	• Compute capability: 3.0 • Memory bandwidth: 80.0 GB/s • Power usage: 68 W
CPU	Intel core i7 4790 [393] • 4 cores, 8 threads • 3.60 GHz (4.00 GHz turbo boost) • TDP: 84 W
RAM	16 GB

***Table 9.2*. Computer software details.**

Component	Details
Operating system	Windows 10 Enterprise
Methodology software	Matlab R2018a [394] • Parallel computing toolbox • Statistics and machine learning toolbox • Deep learning toolbox • Computer vision toolbox • Curve fitting toolbox • Image acquisition toolbox • Image processing toolbox • Robotics systems toolbox • Signal processing toolbox
Development tools	Magician Studio [395] Arduino IDE v1.8.12 USBlyzer (USB protocol analyser and traffic sniffer)

The control box housed the power and control systems. AC power was converted to appropriate DC voltages. 24 V was used to power the stepper motor driver and LEDs. 5 V was used to power the microcontroller. Details regarding the componentry within the control box are provided in Table 9.3.

***Table 9.3*. Control box details.**

Component	Details
AC-DC PSU (24V)	Mean Well MDR-60-24 [396] • Input: 100-240 VAC, 1.8 A • Output: 24 V, 2.5 A
AC-DC PSU (5V)	Chinfa AMR1-05 [397] • Input: 90-264 VAC • Output: 5 V, 1.5 A
Stepper motor driver	Leadshine M542 [398] • Input: 20-50 VDC • Output: 4.2 A max per phase • Microstep resolution: 2, 4, 8, 16, 32, 34, 128
Microcontroller	Arduino Uno REV3 • Input: 5-20 VDC • Digital I/O pins: 14 • Analog pins: 6 • PWM pins: 6 • Clock speed: 16 MHz

9.2.1 Conveyor system

The conveyor system was driven by a NEMA 23 stepper motor directly coupled to an aluminium roller, fixed with bearings. The belt was matt white and moved over an aluminium base, with two plastic guard rails to guide objects entering the vision system away from the edges. The conveyor system rested on 4 load-cells. Figure 9.5 illustrates the conveyor system.

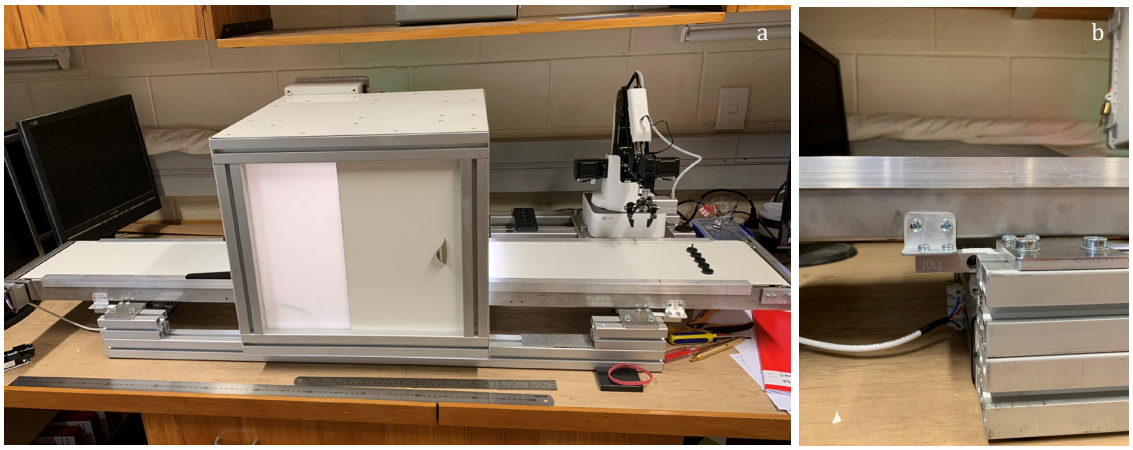


***Figure 9.5*. Illustration of the conveyor system. (a)—front view of the constructed prototype. (b)—10 kg load-cell used to find the COG of an object.**

Shielded cabling was used to transmit data from the load-cell amplifiers to the microcontroller. Note that an amplifier measures and converts the analogue signal from the load-cell to digital data. Consequently, no noise was introduced when communicating the signal from amplifier to microcontroller. The output signal is measured by a Wheatstone Bridge within the load-cell, which responds with a change in voltage as weight is applied to the component. For more information regarding this type of strain gauge configuration, the reader is referred to the resource by Hoffmann [399]. Amplifiers were located directly under each load-cell to minimise the length of wiring, thereby reducing the amount of voltage induced by the AC power system. This resulted in significant noise reduction. The cabling used between amplifiers and load-cells was also shielded. Table 9.4 provides information regarding the specific componentry used by the conveyor system.

***Table 9.4*. Details of componentry utilised in the conveyor subsystem.**

Component	Details
Load-cells	TAL220 Parallel beam load-cell [400] • Capacity: 10 kg • Rated error: ± 0.05%
ADC amplifier	HX711 ADC [401] • Input: 2.7-5.5 VDC • Selectable gain: 32, 64, 128 • 24-bit, serial output • 80 samples per second max
Stepper motor	NEMA 23 SY57STH76-2804A [402] • Single-step angle: 1.8° (200 steps/revolution) • Holding torque: 19 kg-cm • Power: 3.2 V, 2.8 A per phase

ADC amplifier prescale calibration values were derived through experimentation. Each load-cell unit was calibrated individually. Note that a load-cell unit consists of an HX711 amplifier and TAL220 load-cell pairing. Calibration was specific to each load-cell unit and varied slightly between units. 50 initial measurements were taken with no weight present to tare the unit. A calibration tool was bolted to the load-cell and used for calibration. The calibration tool and bolts weighed 1079.60 g, determined by a lab scale certified to within 0.01 g. A total of 20 weight measurements were captured and averaged per trial. Calibration values were adjusted until this trial value was consistent with the calibration tool weight. The returned load-cell unit measurement accuracy was initially set to 0.01 g for calibration, although measurements were generally too noisy at this level of accuracy. Despite the rated error of $\pm$ 0.05% (5 g) cited by the manufacturer, measurement accuracy was found to be consistent within $\pm$ 0.1 g of the calibration weight for individual load-cell units. Once all individual load-cell units were calibrated, the conveyor unit was fixed and new tare values were found. A 133.63 g calibration disc was used to measure the combined load-cell error. Table 9.5 tabulates the errors associated with the weight measurement systems of the conveyor.

***Table 9.5*. Error and calibration tool weights associated with the load-cell-conveyor subsystem. The combined load-cell error was derived from 20 measurements using the calibration disc weight.**

Component	Associated error
Combined load-cell error	$\pm$ 0.10 g
Calibration tool weight	1079.60 g $\pm$ 0.01 g
Calibration disc weight	133.63 g $\pm$ 0.01 g

A NEMA 23 stepper motor was driven in full-step mode, i.e. 200 steps per revolution. Incremental holding torque is theoretically the highest when driving a stepper motor in single steps. Figure 9.6 illustrates the relationship between holding torque and step resolution. Microstepping provides higher step resolution at the cost of holding torque and may result in step loss, thus affecting overall accuracy. This is only applicable however, if this accuracy is reliant on a resultant motor step from a step signal, and can be avoided by implementing alternate means of measuring motor position. M. Walter offers a comprehensive article related to this topic [403].

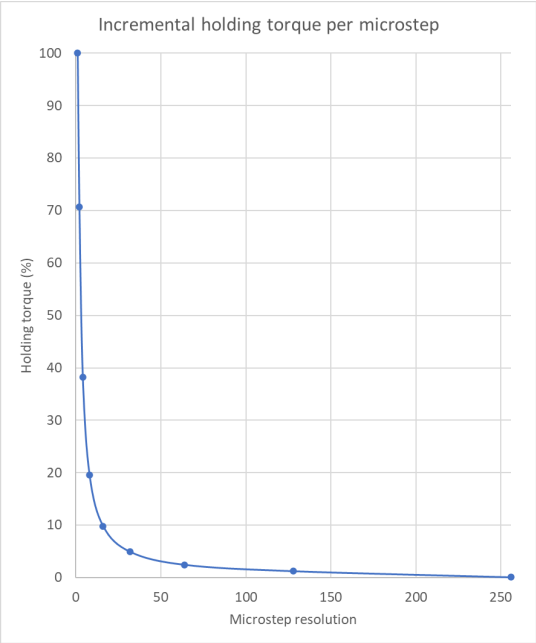


***Figure 9.6*. Graphed theoretical incremental holding torque for step resolutions: 1, 2, 4, 8, 16, 32, 64, 128, 256. Source: MICROMO Electronics Inc. [404].**

The conveyor belt translational accuracy provided by the stepper motor was measured by fixing a sharp, lightweight object to the belt. The object was translated by 10,000 steps. This change in location was measured to the nearest millimetre, repeated 20 times. The resultant step resolution was found to translate the conveyor belt by approximately 0.1 mm per step. A linear speed controller was implemented to smooth stepper motor acceleration and deceleration. Atmel provide an in-depth, mathematical description of this control [405]. Stepper control was implemented in Arduino using an interrupt service routine. Full code is provided online. A link can be found at the end of Section 11.2. Details regarding the stepper and consequent translational properties are provided in Table 9.6.

***Table 9.6*. Error associated with the stepper-conveyor subsystem.**

Component	Associated error
10,000 steps	957 mm $\pm$ 2 mm
Steps per mm	10.45 steps/mm $\pm$ 0.02 steps/mm
Single step	0.0959 mm $\pm$ 0.0002 mm
Max speed	9,615 steps/s
	920 mm/s
	48.1 RPM
Operating speed	5,155 steps/s
	492 mm/s
	25.8 RMP

Due to inaccuracies in construction, the conveyor platform surface was not perfectly perpendicular with the robotic manipulator z-plane. This offset is illustrated in Figure 9.7.

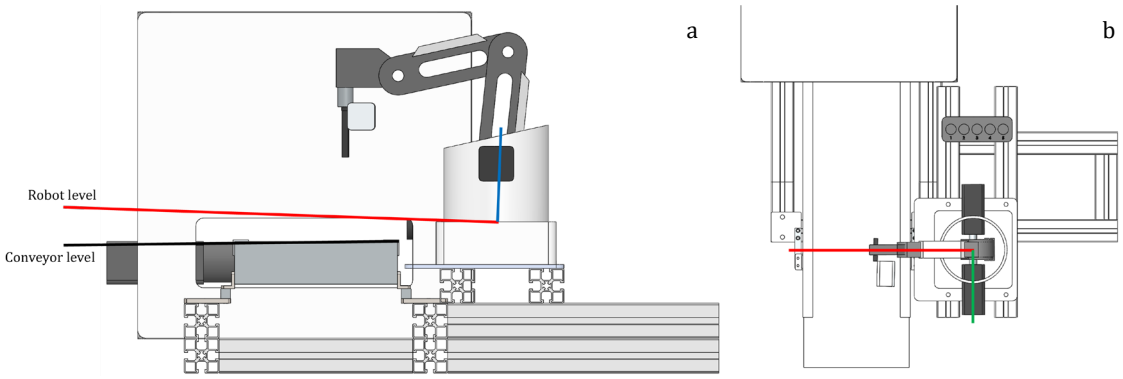


Figure 9.7. **Illustration of misalignment due to inaccuracies in construction. (a)—side view illustrating offset. Note this offset has been exaggerated for clarity. (b)—top view illustrating relative position of conveyor level and robot plane. Red, green and blue express the x-, y- and z-axes, respectively.**

To compensate for this error, bed level was modelled in terms of robot z-axis values. This was achieved by sampling end-effector z-axis positions of the conveyor surface along the x-axis with y-axis coordinates set to 0. Prototype coordinate frames are illustrated in Figure 4.4, Chapter 4.2. A second-degree polynomial relationship was found to relate the true conveyor bed height with the robot x-axis, illustrated in Figure 9.8. This model was used to mitigate the slight error between the robot z-plane and conveyor level surface (Figure 9.7), improving manipulation accuracy.

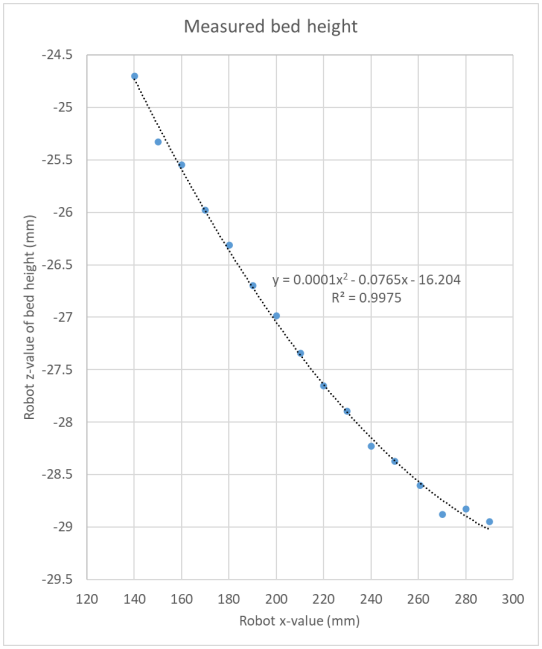


Figure 9.8. **Measured conveyor level height in terms of robot z-value along the robot x-axis. This relationship was used to mitigate misalignment errors due to inaccuracies in construction.**

9.2.2 Vision enclosure

The vision enclosure utilised in this work employed two HD webcams, diffuse LED lighting and photoelectric through-beam sensors. Aluminium sheets shielded the enclosure from external light. Sheet internals were matt white. A sliding door was included at the front of the enclosure for ease of access. Various images related to the enclosure are shown in Figure 9.9.

***Figure 9.9*. Various images related to the vision enclosure of the constructed prototype. (a)—frontal view of the complete enclosure. (b)—internal view showing relative camera placement. (c)—internal view showing cameras and beam sensor.**

Two identical Microsoft Livecam Studio webcams were used for vision—mounted as per the proposed methodology outlined in Chapter 4. Specifications related to the components present in the vision enclosure are tabulated in Table 9.7.

***Table 9.7*. Details of componentry utilised in the vision enclosure subsystem.**

Component	Details
Cameras	Microsoft Livecam Studio [406] • Sensor resolution: 1920 × 1080 • CMOS sensor • 75° diagonal field of view • Auto focus: 0.1 m to ≥ 10 m • 30 FPS • Automatic image adjustment (with manual override) • Digital pan, tilt, zoom • Integrated microphone
LED strips	RS PRO 136-3579 white LED strips [407] • Input voltage: 24 VDC • Protective rating: IP20 • Colour temperature: 6500 K • Power usage: 14 W
Light diffusor	Frosted acrylic sheet
Beam sensor	Photoelectric diffuse-reflective sensor PA18C [408] • Input voltage: 10-30 VDC • Protective rating: IP67 • Response time: ≤ 1.0 ms • Range: 1 m • Operating frequency: 500 Hz

The webcams employed were well-suited to the application. Their relatively low-cost and high inherent sensor resolution provided a cost-effective, high definition solution to vision. All automatic settings on this model could be overridden manually. This feature significantly helped to maintain a consistent inspection environment during development and between trials. Camera settings are shown in Table 9.8.

***Table 9.8*. Internal camera control properties set in Matlab.**

Property	Side-view camera	Top-view camera
Resolution	1920 × 1080	1920 × 1080
ExposureMode	'manual'	'manual'
Brightness	140	140
Contrast	5	5
Exposure	-11	-11
FocusMode	'manual'	'manual'
Focus	25	12
Saturation	100	100
Sharpness	25	25
WhiteBalanceMode	'manual'	'auto'
WhiteBalance	3000	-
BacklightCompensation	0	-

Property values were initially derived from the camera, automatically. These values were then fixed manually. Note that the WhiteBalanceMode was set to 'auto' for the top-view camera. 2 RS PRO white LED strips mounted above a frosted acrylic sheet were used to illuminate the enclosure. LED lighting is generally considered ideal for most applications due to the associated life expectancy, output stability and cost [224, 409]. The diffusor improved inspection quality by reducing harsh shadows from direct light. An emitter and receiver pair of photoelectric beam sensors were fixed at the entrance of the enclosure, i.e. the end furthest from the robotic manipulator. Beam sensors were used to detect and centre new objects to the mid-point of the vision enclosure. For more information regarding this process, refer to Section 9.3.5. Alternatively, objects may be detected and translated via vision. Beam sensors were preferred due to their high sampling rate of 500 Hz, compared to the 30 Hz sampling rate of the webcams.

9.2.3 Robotic manipulator

The robotic manipulator used in this research was part of the Dobot Magician pedagogical package [410]. This product is suitable for hobbyists and education, although the company that manufactures this system also specialises in industrial robotics. The system was relatively low-cost and provided ample features required for this work. Figure 9.10 illustrates the robotic manipulator used in this study.

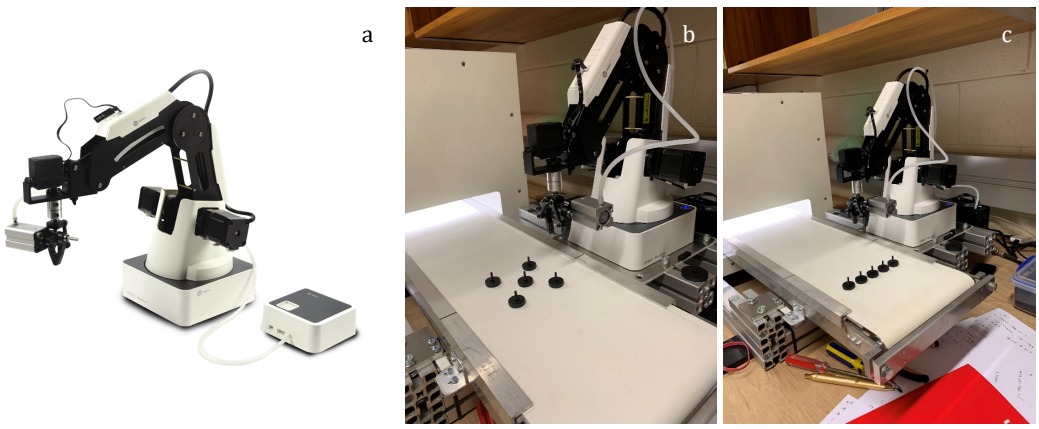


***Figure 9.10*. (a)—Dobot Magician manipulator stock image. Image source: Dobot Magician website [410]. (b), (c)—various images of the Dobot Magician manipulator fixed to the prototype.**

The Dobot Magician was rated to carry a maximum payload of 500 g, while maintaining a position repeatability of 0.2 mm up to 320 mm from the base. Several end-effectors were included in this package. A programmable I/O interface was present on the forearm and base of the robot to encourage the development of custom end-effectors and peripheral hardware. A comprehensive specification list is tabulated in Table 9.9.

***Table 9.9*. Dobot Magician operational specifications. Source: Dobot Magician User Guide V1.5.1 [411].**

Component	Details
Robot Model	Dobot Magician [410]
Maximum payload	500 g
Maximum reach	320 mm
Number of axes	4
Motion range	Base: $[-90°, 90°]$
	Read arm: $[0°, 85°]$
	Forearm: $[-10°, -90°]$
	End-effector rotation: $[-135°, 135°]$
Maximum motion speed (250 g payload)	Base: 320 °/s
	Read arm: 320 °/s
	Forearm: 320 °/s
	End-effector rotation: 480 °/s
Position repeatability	0.2 mm
Power	Input: 12 VDC, 7 A
	Consumption: 60 W max
Physical	Net weight: 3.4 kg
	Footprint: 158 × 158 mm
End-effectors	3D printer kit
	Laser engraver
	Pen holder
	Vacuum suction cup
	2-fingered pneumatic gripper • Range: 0-27.5 mm • Maximum force: 8 N • Payload: 500 g
Software	DobotStudio
	DobotBlocky
	Proprietary communication protocol
	Dobot program library
Extensions	10 × I/O interfaces
	4 × Controllable 12 V power output
	UART communication interface
Communications	USB / Wi-Fi / Bluetooth

Robot kinematic calculations were handled by an onboard controller embedded in the base of the manipulator. Stepper motor drivers and the related control electronics were also housed in the base. Robot configuration could be requested via USB, Wi-Fi or Bluetooth in terms of a joint coordinate system or mm-based Cartesian coordinates relative to the end-effector, shown in Figure 9.11. The manipulator was stand-alone, in that only a power supply and top-level control signal were needed for operation. A diagram depicting manipulator workspace is provided in Figure 9.12.

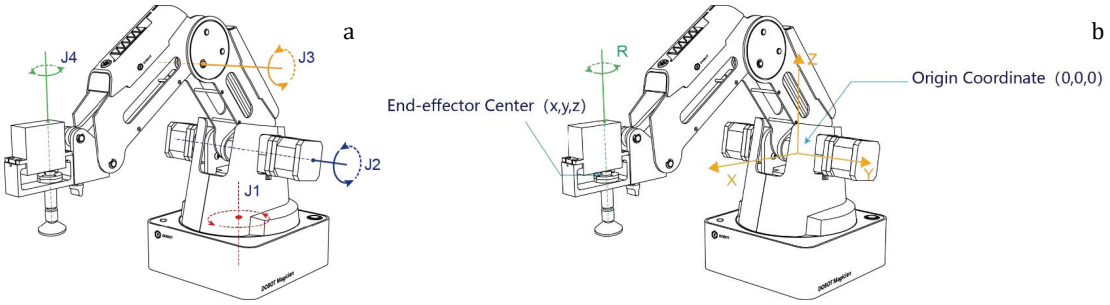


Figure 9.11. **(a)—joint coordinate system. (b)—Cartesian coordinate system. Image source: Dobot Magician User Guide V1.5.1 [411].**

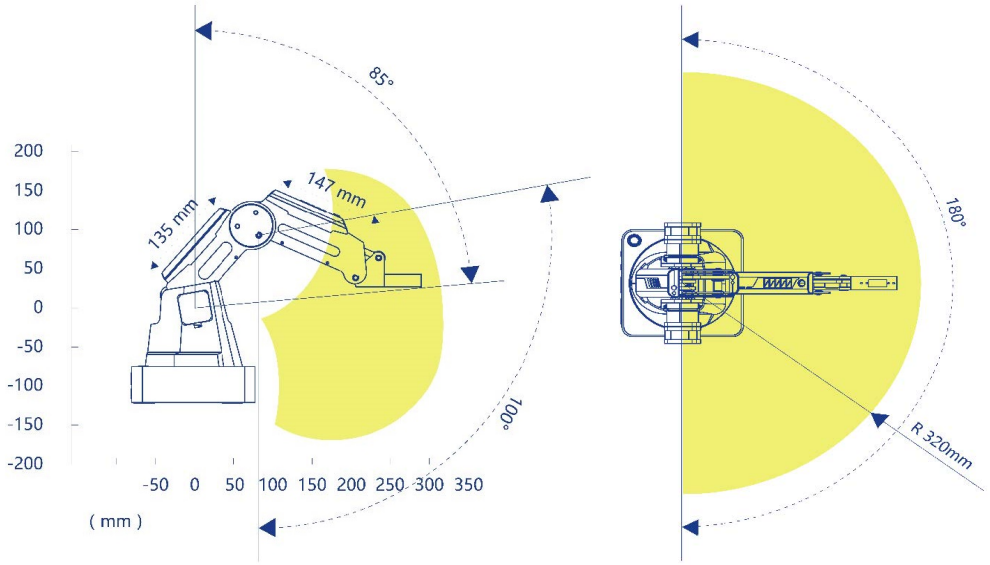


Figure 9.12. **Rated Dobot Magician workspace. Image source: Dobot Magician User Guide V1.5.1 [411].**

DobotStudio was the proprietary software that accompanied the manipulator. This software provided an environment for 3D printing, laser engraving, leap motion, troubleshooting, firmware updates and GUI manipulator control. Moreover, control could be simplified via program blocks, direct script, write & draw and teaching & playback. Note that an unlock button was present on the forearm of the manipulator. This disabled all stepper motors, while continuing to read arm positional encoders—thus, robot configuration could be set by the user directly.

To streamline the control pipeline, commands to the manipulator microcontroller were sent directly from Matlab through serial communication via USB. To aid developers, Dobot released a custom Matlab API. Version 1.0 was released on 30 June, 2018 [412]. Installation of this library was very difficult, and the resulting control was slow, prone to error and extremely limited in terms of functionality. Note that an update was released on 29 September, 2019. Control of this manipulator was further confounded by the non-standard communication protocol implemented by the manufacturer. Fortunately, their protocol is open-source and available on their website. This work makes use of the Dobot Magician Communication Protocol V1.1.3 document [413], 16 November, 2018 release. Unfortunately, this document was incomplete and error-prone—making it difficult to develop a serial control algorithm. Consequently, the USB protocol analyser and traffic sniffer noted in Table 9.2, Section 9.2 was used to intercept and record command packets sent to the manipulator from DobotStudio. A combination of the protocol document and

intercepted commands were used to write a custom library to facilitate efficient, robust and timely control of the manipulator from Matlab. An interpretation of Magician command protocol is available in an online article published by the author of this thesis [414].

9.3 Vision

Prior to digital image processing, lens distortion parameters were derived through calibration. The Matlab Single Camera Calibrator App was used with the default checkerboard pattern provided. Figure 9.13—a illustrates this pattern.

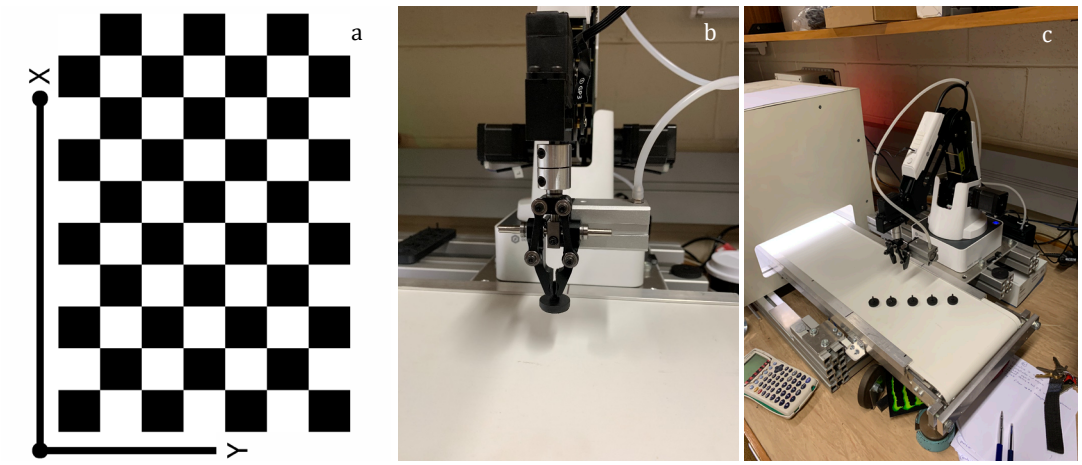


Figure 9.13. **(a)—Matlab checkerboard camera calibration pattern. Image source: Matlab computer vision Toolbox [394]. (b)—Dobot Magician picking a 20 mm calibration disc. (c)—5-disc calibration pattern used to consolidate robot and vision coordinate systems.**

No significant lens distortion was found. Therefore, both cameras were operated without this correction to save computation time. 5 calibration discs with rubber bottoms were used to tune the vision system, find threshold values and consolidate coordinate systems, shown in Figure 9.13—c. Discs were matt black with 20 mm diameters. An approximate pixel to mm conversion was found by calculating the disc area in mm:

$$\mathrm{A}_{disc_{mm}} = \frac{\pi D^2}{4} \tag{9.1}$$

where $\mathrm{A}_{disc_{mm}}$ is the calculated area of a calibration disc from a top-down perspective in squared millimetres. Note that diameter D was measured with a digital micrometer to within 0.01 mm. A conversion may then be established through relationship:

$$mm\ per\ pixel = \sqrt{\frac{\mathrm{A}_{disc_{mm}}}{\mathrm{A}_{disc_{pix}}}} \tag{9.2}$$

where $\mathrm{A}_{disc_{pix}}$ is the area of the disc measured by the vision system, in pixels. This calculation provides an approximate conversion from mm to pixel. For a conversion from pixels to mm, the reciprocal may be calculated as:

$$pixels\ per\ mm = \frac{1}{\sqrt{\frac{A_{disc_{mm}}}{A_{disc_{pix}}}}} \tag{9.3}$$

Note that this conversion is only an estimate and assumes a linear relationship between pixel and mm. Due to minor lens distortion, this value may vary slightly between x-axis, y-axis and distance from the camera. This tentative value, along with threshold and other vision-related parameters used in grasp pose generation is provided in Table 9.10.

***Table 9.10*. Implemented variable values associated with the grasp pose generation and selection methodology described in Chapter 4.**

Variable	Description	Value
H	Height of top-view image space I_t and side-view image space I_s.	1080
W	Width of top-view image space I_t and side-view image space I_s.	1920
h_t	Height of rectangular grasping window I_{RGW} and classification window I_{RCW}. Height of stage 2 input samples.	52
w_t	Width of rectangular grasping window I_{RGW} and classification window I_{RCW}. Width of stage 2 input samples.	164
T_t	Threshold used to create binary image $I_{t_{Th}}$ from I_t.	191
$H_{I_{BB}}$	Classification bounding box I_{BB} height and width. Downscaled to 198 for classification. Height and width of stage 1 input samples.	790
T_{class}	Classification acceptance threshold. Value used to threshold stage 1 softmax output.	-
T_{grasp}	Grasp hypothesis acceptance threshold. Value used to threshold stage 2 softmax output $K_{I_{RGW}}$.	0.7
i_{step}	Horizontal step magnitude in rotated space $I_{BBR_{1\ldots4}}[i,j]$.	10
j_{step}	Vertical step magnitude in rotated space $I_{BBR_{1\ldots4}}[i,j]$.	20
$y_{s_{step}}$	Vertical step magnitude in side-view image space I_s.	20
T_{I_s}	Threshold used to create side binary image $I_{s_{Th}}$ from $I_{s_{diff}}$.	217
$pixels\ per\ mm$	Number of pixels per mm (I_t).	6.6191

Threshold values were found through experimentation. Fixed global thresholds for binary segmentation could be applied robustly due to the consistency of the conveyor material, lighting and camera settings. Global thresholds are discussed in Chapter 2.2.

The 2-fingered gripper attachment employed by the Dobot Magician system had a maximum opening of 27.5 mm and width of 8 mm. Rubber inserts were added to the inside of the gripper, reducing the maximum opening to 25 mm. This area corresponds to the rectangular grasping window I_{RGW} in top-view image space I_t. I_{RGW} dimensions were found by 3D printing a 25 × 8 mm rectangular calibration part. This part was imaged at various locations within I_t, see Figure 9.14—b. The height h_t and width w_t of this part were measured as 164 pixels and 52 pixels, respectively.

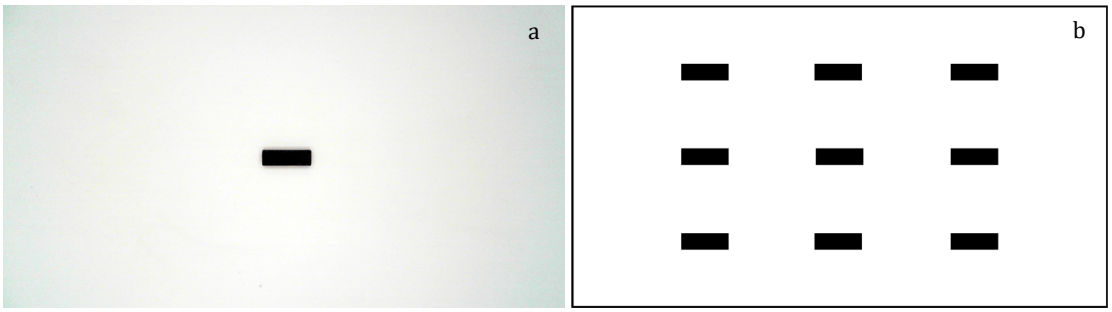


Figure 9.14. **(a)—top-view image I_t containing 3D printed calibration part dimensioned according to the implemented gripper. (b)—calibration pattern used to find I_{RGW} dimensions in pixels.**

To help facilitate pragmatic implementation of the proposed methodology, all coordinate frames present in the prototype were consolidated with respect to the top-view vision coordinate frame I_t, e.g., robot coordinate space, side-view image space I_s and load-cell space.

9.3.1 Conveyor belt coordinate frame consolidation

The conveyor belt was actuated in one axis only. The top-view I_t x-axis and conveyor coordinate frame act in the same plane—graphically depicted in Figure 9.15. Figure 4.4, Chapter 4.2 provides a more complete illustration of the coordinate frames associated with the proposed prototype.

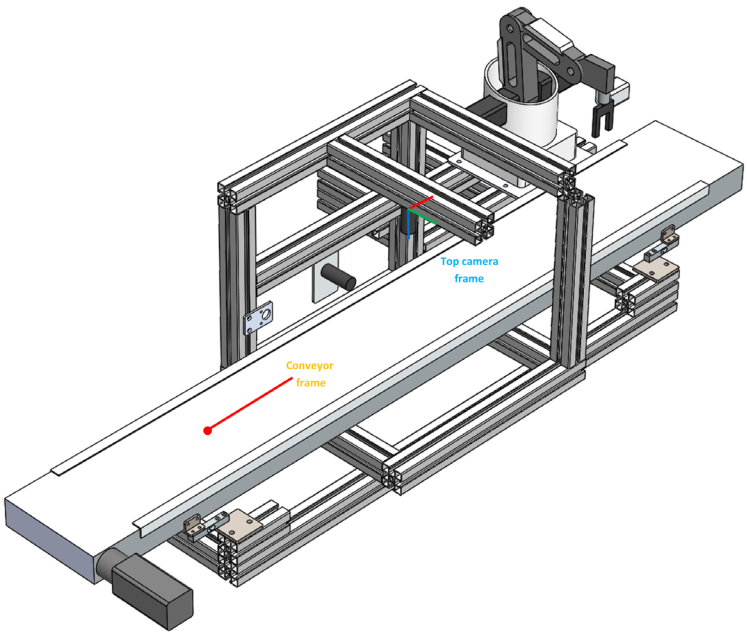


Figure 9.15. **Visualisation of the conveyor belt coordinate frame with respect to top-view vision frame I_t. The top-view x-axis and conveyor frame act in the same plane. Red, green and blue express the x-, y- and z-axes, respectively.**

The conveyor coordinate system is expressed in terms of motor steps. One step corresponds to approximately 0.096 mm, or 0.64 pixels. For more information, see Table 9.6, Section 9.2.1. The only function of the conveyor is to move objects in and out of coordinate frames. Initially, objects placed on the conveyor belt are translated to an approximate x-axis mid position within top-view image space I_t by a beam sensor process. Section 9.3.5 details this procedure. At this stage, objects are discoverable by I_t and I_s, therefore a grasp configuration may be calculated. Top-view image space I_t and robot workspace $\bar{G}$ may be

treated as though they occupy the same space by translating the object a set amount between frames. Effectively, conveyor space is ignored and treated as an intermediary consolidation step prior to an interaction. Figure 9.16 illustrates the function of the conveyor system.

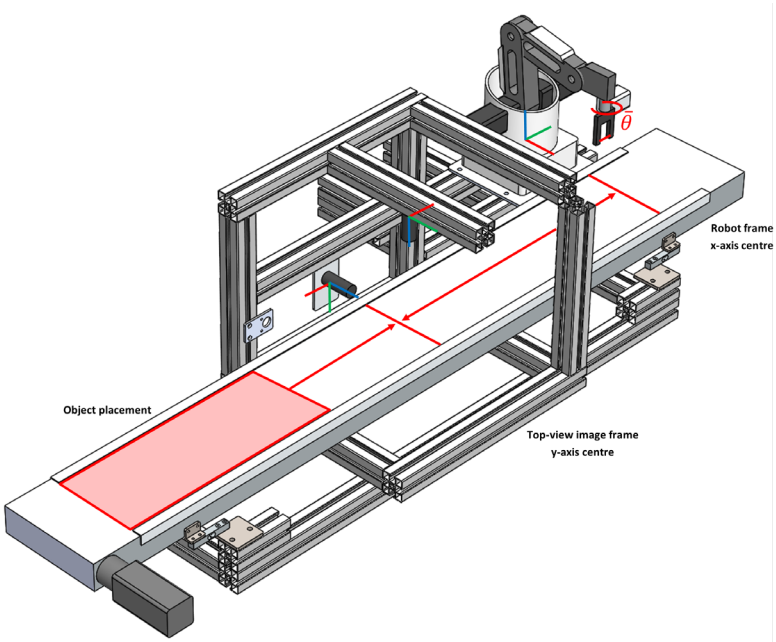


***Figure 9.16*. Illustration of the function of the conveyor system. Objects are placed haphazardly at one end of the prototype, where they may be moved into the desired coordinate frame by the conveyor. Red, green and blue coordinate lines express the x, y and z-axes, respectively.**

Although image space I_t and robot workspace $\bar{G}$ can be treated as though they overlap, these frames must still be consolidated. Note that the intermediate conveyor translate step introduces a fixed error within this transformation. The error associated with the conveyor belt was measured within I_t by translating calibration discs various step amounts and recording the resultant offset. This error is noted in Table 9.11.

***Table 9.11*. Error associated with the conveyor system, derived from 50 measurements.**

Component	Associated error
Conveyor belt translational error (pixels)	$\pm$ 3.4 pixels
Conveyor belt translational error (mm)	$\pm$ 0.5 mm

The measurements cited in Table 9.11 relate to errors in conveyor translation accuracy and axis misalignment, e.g. the top-view I_t x-axis and conveyor axis may not perfectly align. The distance from the end of the beam sensor to I_t x-axis centre was 1,640 steps (157 mm).The distance from I_t x-axis centre to $\bar{G}$ x-axis zero was 5737 steps (549 mm).

9.3.2 Robot coordinate frame consolidation

Although the robot coordinate frame $\bar{G}[\bar{x}, \bar{y}, \bar{z}, \bar{\theta}]$ is significantly offset from the top-view camera frame I_t, many axes act in the same direction. Figure 9.17 illustrates the relative origin positions of these coordinate spaces. The Dobot Magician Cartesian coordinate system defines the position of the end-effector relative to an origin situated inside the base of the robot. Figure 9.11—b, Section 9.2.3 graphically depicts the Cartesian coordinate

frame used by the Magician manipulator. Note that the Magician workspace is defined in terms of millimetres.

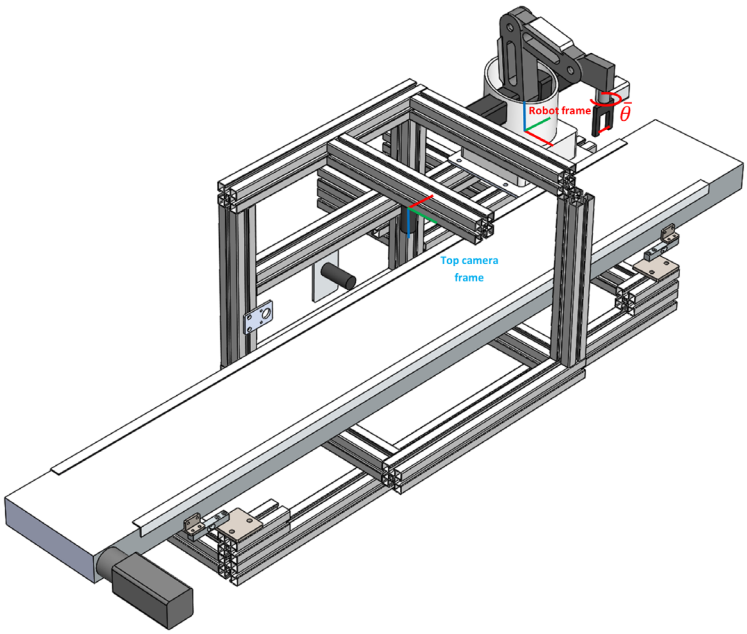


Figure 9.17. **Illustration of the robot coordinate frame $\bar{G}[\bar{x}, \bar{y}, \bar{z}, \bar{\theta}]$ with respect to the top-view vision frame I_t. Red, green and blue express the x-, y- and z-axes, respectively.**

5 calibration discs were used to consolidate the $\bar{G}[\bar{x}, \bar{y}]$ components of the robot coordinate frame with the top-view camera frame $I_t[x, y]$. The calibration pattern in shown in Figure 9.18.

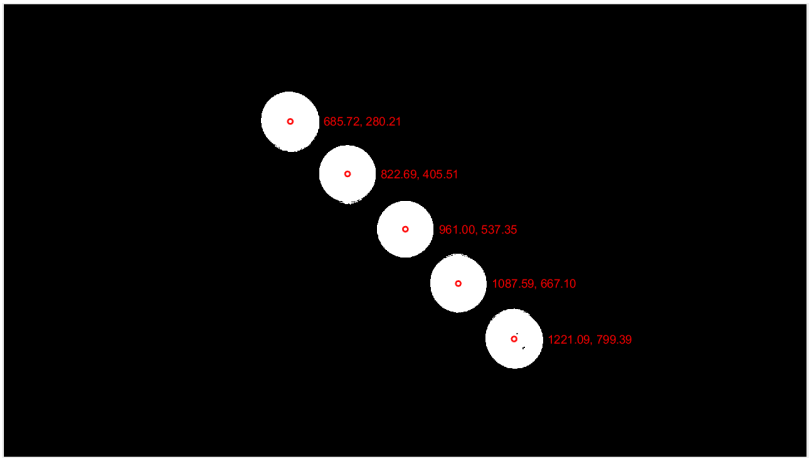


Figure 9.18. **Calibration pattern used to consolidate robot and vision coordinate frames. Vision coordinates are annotated.**

A centre disc is placed at $\bar{G}[\bar{x}, \bar{y}] = [214.4,0]$, coordinates that roughly correspond to the centre of I_t, i.e. $I_t[x, y] = \left[\frac{W}{2}, \frac{H}{2}\right] = [960,540]$. 4 additional discs are placed at even horizontal and vertical offsets in robot space. Once 5 calibration discs were placed by the manipulator, the conveyor system translated the pattern into I_t. Disc centre positions were found through vision. Figure 9.19 graphically illustrates the relationship between $I_t[x, y]$ and $\bar{G}[\bar{x}, \bar{y}]$.

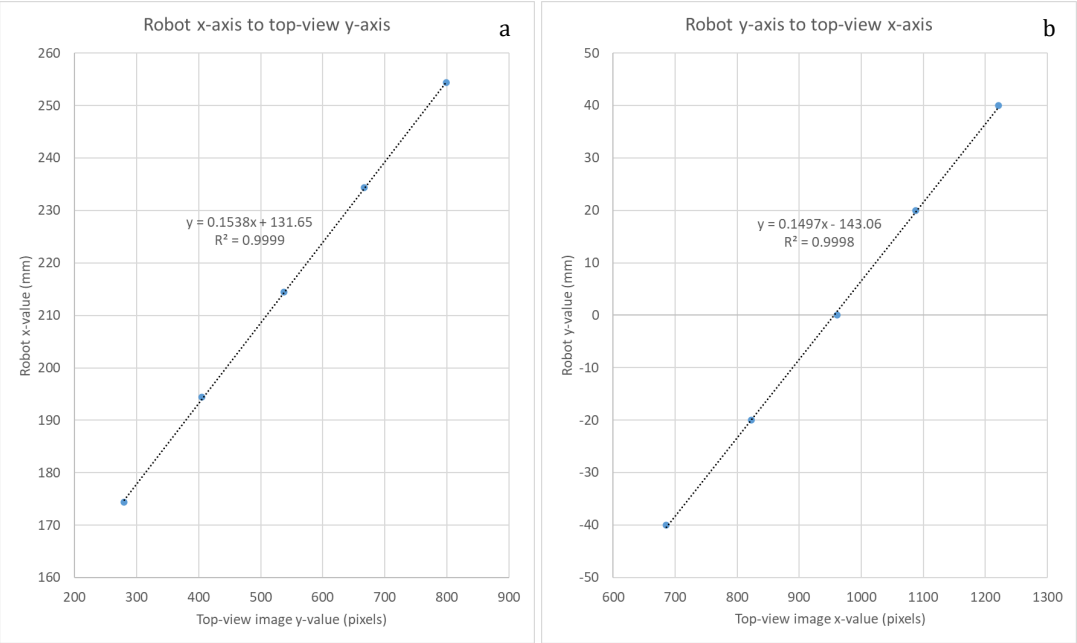


***Figure 9.19*. (a)—relationship between robot x-axis and top-view vision y-axis. (b)—relationship between robot y-axis and top-view vision x-axis. Relationships were measured by the calibration pattern using 20 mm discs. These relationships were used to consolidate robot and vision coordinate systems.**

Linear models were fit to the resulting data. These relationships could be used to predict robot coordinates $\bar{G}[\bar{x}, \bar{y}]$, given top-view camera coordinates $I_t[x, y]$. Therefore, selected grasp candidates $G[x_t, y_t, y_s, \theta_t]$ in image-space, can be converted to robot space $\bar{G}[\bar{x}, \bar{y}, \bar{z}, \bar{\theta}]$ via the following transforms that define tr_{RC}:

$$\bar{x} = 0.15\left(y_t - y_{t_{offset}}\right) + 131.65 \tag{9.4}$$

$$\bar{y} = 0.15\left(x_t - x_{t_{offset}}\right) - 143.06 \tag{9.5}$$

$$\bar{\theta} = \theta_t + 1.72 \tag{9.6}$$

For a comprehensive description of the grasping rectangle representation used in this dissertation, see Chapter 4. Robot end-effector angle $\bar{\theta}$ and top-view grasping rectangle angle θ_t were related by a fixed offset, derived through experimentation. Because the employed 2-fingered gripper is symmetric around $\pm 90°$, $\bar{\theta}$ ranges from $[0°, 90°]$. The $\bar{z}$ component of the tr_{RC} transform is discussed in Section 9.3.3.

The encoder system utilised by the Dobot Magician package is defined relative to some local home position. This position is lost when powered down. Therefore, the manipulator must be initialised prior to operation, a process which defines a new coordinate origin. Unfortunately, this origin varied by approximately 2 mm between initialisations. To mitigate this error, $\bar{G}[\bar{x}, \bar{y}] = [214.4,0]$ and $I_t[x, y] = [960,540]$ were assumed to represent the same point in space. Therefore, the relative pixel offset introduced by a new home position and errors in the point-overlap assumption could be measured in I_t via one calibration disc at each new initialisation. X-axis and y-axis pixel offsets were denoted as $x_{t_{offset}}$ and $y_{t_{offset}}$, respectively. The linear relationships used by transform tr_{RC} were assumed between initialisations.

No significant rotational offset was found between coordinate frames. Therefore, simple linear relationships could be applied accurately, as opposed to computationally expensive rotation matrices that account for rotational offsets between coordinate systems. Post-initialisation, Equation 9.4 and Equation 9.5 could be used to predict $I_t[x, y]$ from known $\bar{G}[\bar{x}, \bar{y}]$ positions. Consequently, calibration discs were used to measure the tr_{RC} transformation error in terms of pixels in I_t. Measurement of this error encapsulates the error associated with the positional repeatability of the robotic manipulator. The discrepancy between the robot and vision coordinate frames is provided in Table 9.12.

***Table 9.12*. Error associated with the vision-robot transform tr_{RC}, found from 50 measurements.**

Component	Associated error
Robot-vision transformation error (pixels)	$\pm$ 1.70 pixels
Robot-vision transformation error (mm)	$\pm$ 0.24 mm

9.3.3 Side-view image space consolidation

The vision system proposed in this thesis utilises two cameras. The top-view camera frame I_t and side-view camera frame I_s image the same space from relatively perpendicular perspectives. Thus, camera frames are offset by 180° and share an x-axis plane—though the direction is inverted. Figure 9.20 illustrates the relative position of the camera coordinate frames.

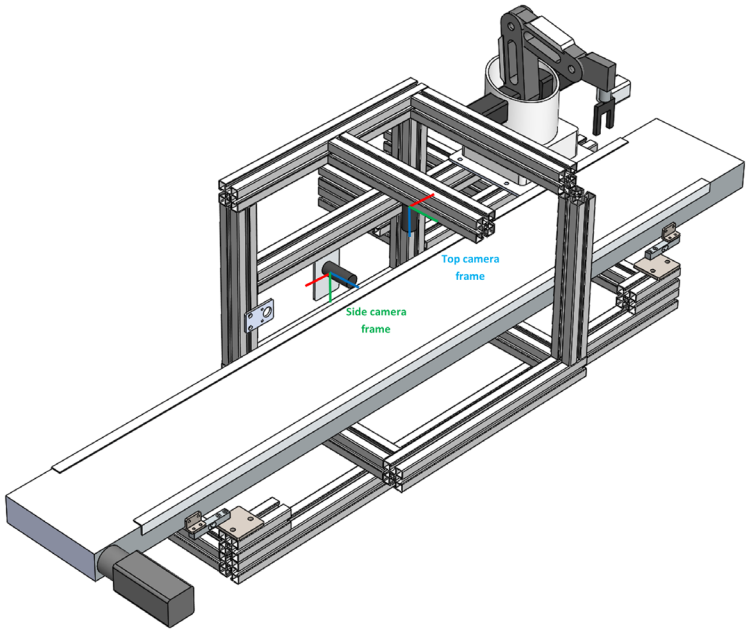


***Figure 9.20*. Illustration of the side-view vision frame I_s with respect to the top-view vision frame I_t. Red, green and blue express the x-, y- and z-axes, respectively.**

As the location of a grasping rectangle $G[x_t, y_t, y_s, \theta_t]$ tends toward the side camera origin, rectangle size from the side-view perspective increases, i.e. as a rectangle position from perspective I_t approaches the side-view coordinate origin, side-view rectangle height h_s and width w_s increase in value. This relationship may be modelled, provided the distance from a rectangle to the side-view camera is known. Given the configuration of the proposed prototype, rectangle-to-side-camera distance can be approximated in pixels by the I_t y-axis. Thus, rectangle dimensions h_s, w_s can be modelled in terms of $I_t[y]$. Refer to Figure 4.6, Chapter 4.2 for a comprehensive illustration of the grasping rectangle representation. To

quantify this relationship, the 3D printed, 25×8 mm rectangular calibration part representing maximal gripper size was used. This part was placed at various I_t y-axis positions along I_t x-axis centre. The resulting 8 mm gripper width was measured within I_s in pixels. Gripper width in terms of pixels, with varying camera distances is graphed in Figure 9.21.

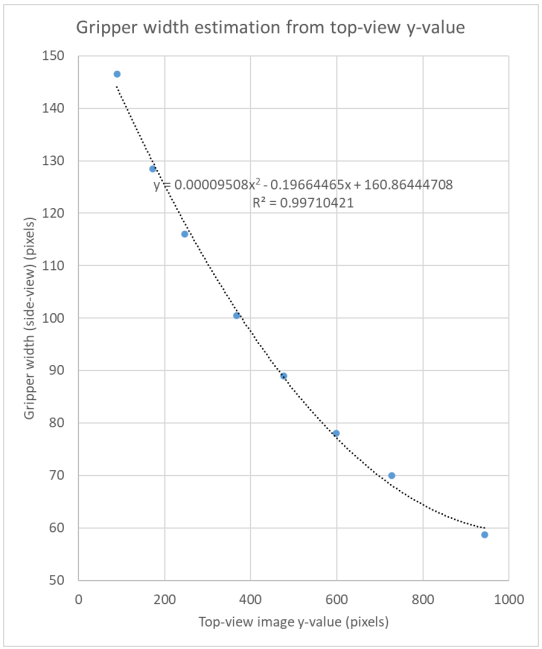


***Figure 9.21*. Relationship between measured gripper width and top-view image y-axis. This relationship may be used to estimate gripper width as a function of the top-view y-axis position.**

The side-view pose component of a grasp $pose_s = \{x_s, y_s, h_s, w_s\}$ is detailed in Chapter 4.2. The resulting polynomial relationship from Figure 9.21 may be used to estimate gripper width in side-view perspective I_s, given top-view grasping rectangle position y_t, thus satisfying the values in transformation tr_{sideW}:

$$w_s = 9.81e^{-5}{y_t}^2 - 0.20y_t + 160.86 \quad \textbf{(9.7)}$$

Gripper height h_s is therefore calculated as:

$$h_s = w_s\left(\frac{h_t}{w_t}\right) = w_s\frac{164}{52} = 3.15w_s \quad \textbf{(9.8)}$$

This relationship relates to the fixed gripper size ratio from the top-view pose component of a grasp $pose_t = \{x_t, y_t, \theta_t, h_t, w_t\}$. Since gripper size is known, the ratio is known.

I_t x-axis and I_s x-axis operate in the same plane but act in opposite directions. Consequently, side-view gripper x-axis pose x_s may be approximated using the top-view gripper x-axis pose x_t. 23 measurements were collected to model this transformation. The rectangular calibration part was placed at top-view y-axis centre $I_t[y] = 540$ and iterated along the top-view x-axis $I_t[x]$. The resulting side-view position $I_s[x]$ was found through vision. Figure 9.22 shows the relationship between top-view x-axis $I_t[x]$ and side-view x-axis $I_s[x]$.

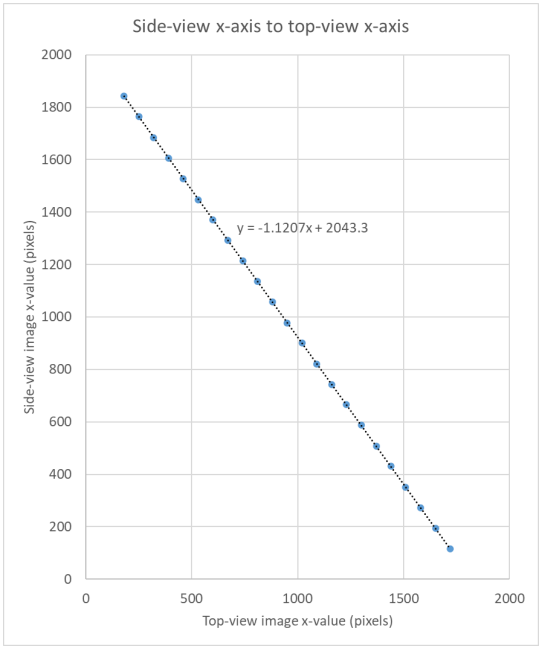


***Figure 9.22*. Measured relationship between side-view image x-axis and top-view image x-axis. This relationship was used to estimate side-view image x-position as a function of the top-view image x-position.**

The relationship graphed in Figure 9.22 satisfies the side-view rectangle x-position x_s in transformation tr_{sidex}:

$$x_s = -1.12x_t + 2043.30 \tag{9.9}$$

This transformation is used to estimate the gripper side-view centre x_s position, given top-view gripper centre x_t. Chapter 4.3 provides information related to the process by which x_t is found. To compute the vertical component of a side-view grasping rectangle y_s, the height of the platform at the point of object-conveyor contact ph_{level} must be known prior. This relationship was modelled as per the method used to find Equation 9.7. The rectangular calibration part was placed at various I_t y-axis positions along I_t x-axis centre $I_t[x] = 960$. The resulting bed height I_s y-axis position was measured. The platform height estimation data is graphed in Figure 9.23.

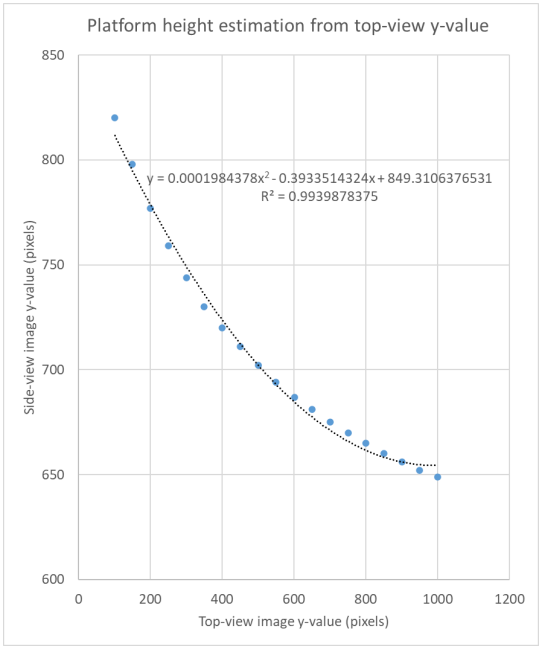


***Figure 9.23*. Relationship between the estimated platform height in terms of the side-view y-axis and the top-view image y-axis. This relationship was used to estimate the platform height as a function of the top-view y-axis position.**

This relationship may be used to calculate platform height ph_{level}, given object top-view y-position y_o:

$$ph_{level} = 1.94e^{-4}{y_o}^2 - 0.39y_o + 849.31 \quad \textbf{(9.10)}$$

Equation 9.10 satisfies the platform level transformation tr_{ph}. Note that platform level is calculated from candidate object y-position y_o, as opposed to grasping rectangle y-position y_t. More accurate estimations of ph_{level} may be achieved by substituting y_o for y_t. However, during experimentation it was found that this slight accuracy increase was insignificant compared to the computational cost of calculating new ph_{level} values with each candidate grasp, which would be the case if y_t was to be used. In contrast, ph_{level} is estimated once per object when using y_o. The vertical component of a side-view grasping rectangle y_s may now be found as per the iterative window process described in Chapter 4.3.

The width of the gripper representation in I_s varies significantly due to perspective. However, this width is fixed at 8 mm. Therefore, an approximate millimetre-to-pixel conversion is defined as:

$$mm\ per\ pixel = \frac{8}{w_s} \quad \textbf{(9.11)}$$

Unlike its top-view counterpart, this conversion changes depending on the gripper width estimation w_s, which is based on the I_t y-axis position of a current grasping rectangle y_t. Therefore, the robot end-effector $\bar{z}$ component of the tr_{RC} transform may be defined as:

$$\bar{z} = bedHeight + \left(ph_{level} - y_s - \frac{h_s}{2}\right)\frac{8}{w_s} \quad \textbf{(9.12)}$$

where:

$$bedHeight = 1.00e^{-4}\bar{x}^2 - 0.08\bar{x} - 16.20 \quad \textbf{(9.13)}$$

$bedHeight$ refers to the robot $\bar{z}$ value of the end-effector at the conveyor surface, i.e. gripper-conveyor point of contact. Bed height varies between -29 and -24, depending on robot x-axis position. See the graph shown in Figure 9.8, Section 9.2.1 for an illustration of this relationship. The I_s y-axis value increases downward—the opposite direction to the robot z-axis $\bar{z}$. Therefore, $\bar{z}$ is calculated with respect to bed height. Note that the Magician $\bar{z}$ value refers to the lowest tip of the end-effector, while y_s effectively denotes the centre position of the gripper from the side-view perspective. As such, an offset of half the gripper height $\frac{h_s}{2}$ is needed when calculating $\bar{z}$. The $\left(ph_{level} - y_s - \frac{h_s}{2}\right)$ component of the $\bar{z}$ calculation is essentially referring to the distance between the bed height and the lowest point of the rectangle representation, in pixels. This value is converted to millimetre and offset from the $bedHeight$ $\bar{z}$ value at the current robot $\bar{x}$ position. The light blue annotation in Figure 9.24 clarifies this calculation.

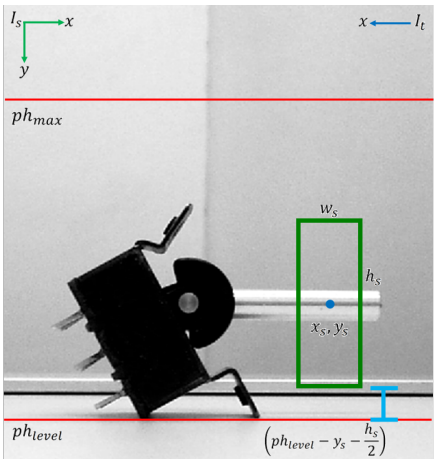


Figure 9.24. **Illustration showing the bed height offset used during the robot $\bar{z}$ calculation.**

9.3.4 Load-cell coordinate frame consolidation

The coordinate frame LC is quantified by 4 sensitive load-cells. The conveyor platform rests on these components, which respond to weight. Load-cell space LC and the top-view camera frame I_t share two axes, though the origin is offset. Figure 9.25 shows the position of each respective coordinate space. Figure 4.12, Chapter 4.2 provides a more comprehensive depiction of the relative coordinate spaces, often referenced in this section.

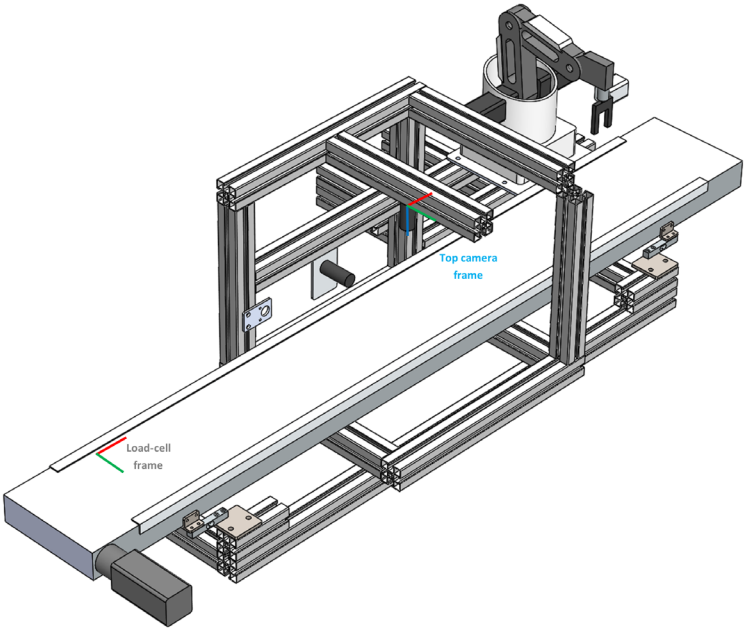


***Figure 9.25*. Illustration of the load-cell coordinate frame *LC* with respect to the top-view camera frame I_t. Red, green and blue express the x-, y- and z-axes, respectively.**

To formalise load-cell coordinate space in terms of x- and y-axes, the output of each component is measured. The combined sum of all load-cell measurements $weight_{total}$ denotes the weight of the object. Equation 4.27, Section 4.2 defines the $weight_{total}$ calculation. Because an object is supported by a fixed number of load-cells—and the total weight is known—the ratio of weight distribution between axes can be calculated. This ratio may be used in conjunction with known distances between load-cells to compute a relative COG position. Patel and Topiwala provide a comprehensive mathematical description of this process [380]. In this work, the ratio coefficients were used to calculate a COG position in I_t space directly—thus, avoiding an additional transform between coordinate frames. Coefficient calculations $coeffx$ and $coeffy$ refer to Equation 4.25 and Equation 4.26, Section 4.2, respectively. A 133.63 g calibration disc was used to measure the response of $coeffx$ and $coeffy$ as the disc was iterated across $I_t[x, y]$. Disc centre location was derived through vision. The resulting relationships between $I_t[x]$ and $coeffx$ and $I_t[y]$ and $coeffy$ are graphed in Figure 9.26.

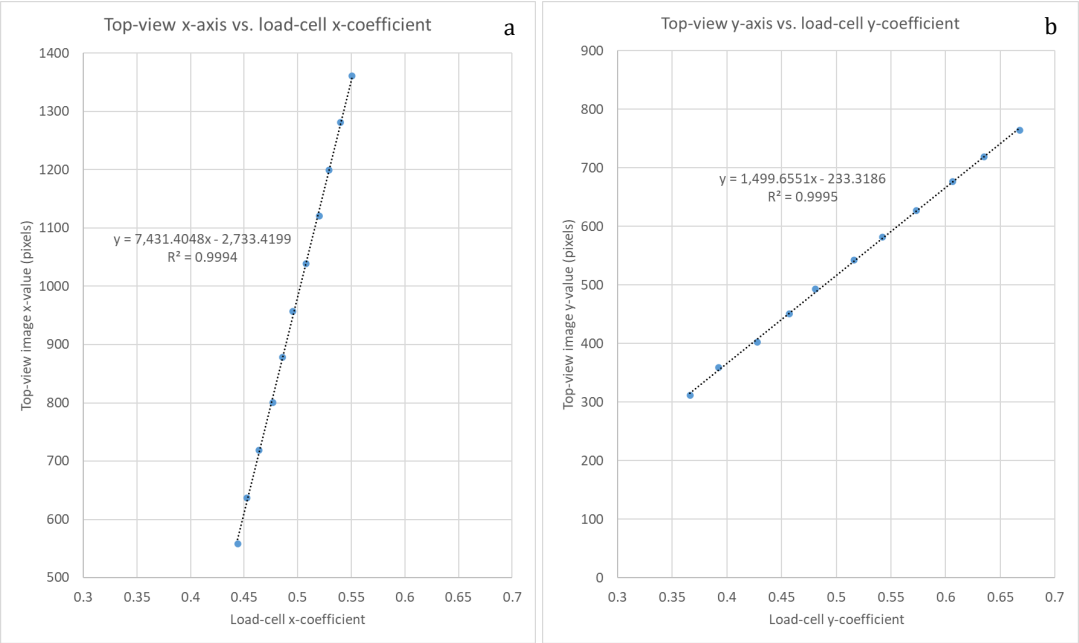


***Figure 9.26*. (a)—relationship between top-view image x-axis and load-cell x-coefficient. (b)—relationship between top-view image y-axis and load-cell y-coefficient. These relationships were used to compute the COG position of an object in top-view image space I_t.**

These relationships satisfy load-cell transforms tr_{LCX} and tr_{LCY}:

$$x_{t_{COG}} = 7431.40coeffx - 2733.42 \quad \textbf{(9.14)}$$

$$y_{t_{COG}} = 1499.66coeffy - 233.32 \quad \textbf{(9.15)}$$

Equations 9.14 and 9.15 were therefore employed to estimate the COG of an object $\left[x_{t_{COG}}, y_{t_{COG}}\right]$ in top-view image space I_t from load-cell measurements directly. Due to the inherent noise associated with the TAL220 parallel beam load-cell and HX711 ADC combination, the accuracy of the resulting system was related to the weight of the object resting on the conveyor, i.e. measurement accuracy increased as object weight increased. This relationship is shown by the examples in Figure 9.27.

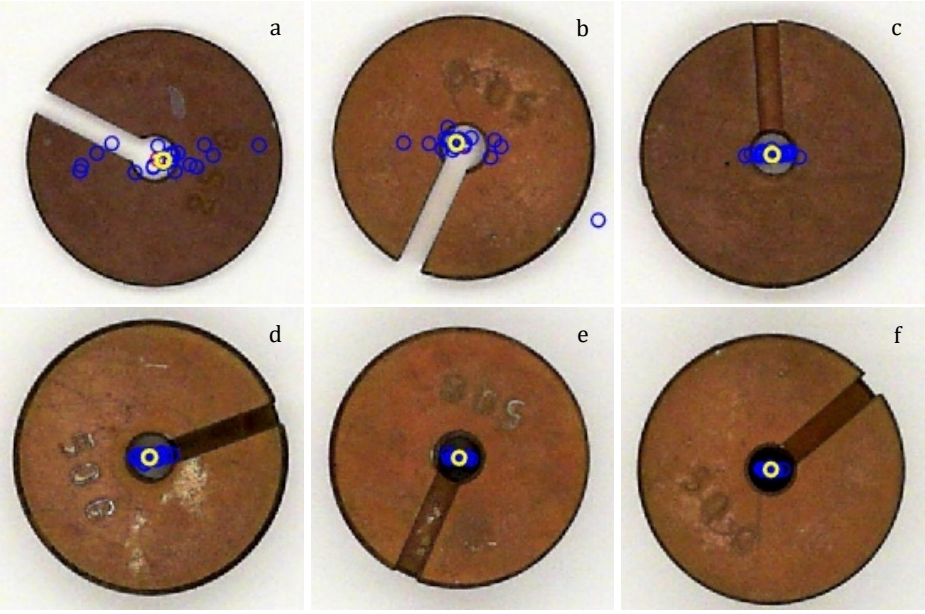


***Figure 9.27*. (a)—25 g weight. (b)—50 g weight. (c)—75 g weight. (d)—100 g weight. (e)—150 g weight. (f)—200 g weight. The computed COG locations for 20 measurements in top-view image space. Blue represents COG measurements. Yellow represents average. Red represents centre of object from vision.**

Load-cell measurements tended to produce stable values when assessing objects weighing at least 75 g. Measurement accuracy improved steadily as weight increased. System accuracy was assessed relative to the top-view camera space I_t. The 133.63 g calibration disc was circular; thus, the object centroid $[x_o, y_o]$ in terms of image space I_t could be found accurately using vision. Refer to Equations 4.30 and 4.31, Section 4.3 for a mathematical description of this process. The centroid position $[x_o, y_o]$ was compared to the COG position $\left[x_{t_{COG}}, y_{t_{COG}}\right]$ of an object in I_t, found via load-cell measurements. System accuracy is provided in Table 9.13.

***Table 9.13*. Error associated with the COG position estimate, taken from 20 measurements using the calibration disc.**

Component	Associated error
COG position error @ 133.63 g (pixels)	$\pm$ 4.1 pixels
COG position error @ 133.63 g (mm)	$\pm$ 0.6 mm

9.3.5 Beam sensor process

Photoelectric diffuse-reflective sensors situated at the entrance to the vision enclosure were used to translate objects to I_t x-axis centre, i.e. $I_t[x] = \left[\frac{W}{2}\right] = [960]$. The pair of beam sensors function by projecting an infrared beam from transmitter to receiver. The receiver unit responds to interruptions in the beam. Therefore, the presence or absence of an object may be determined. Figure 9.28 illustrates the relative position of the beam sensor configuration.

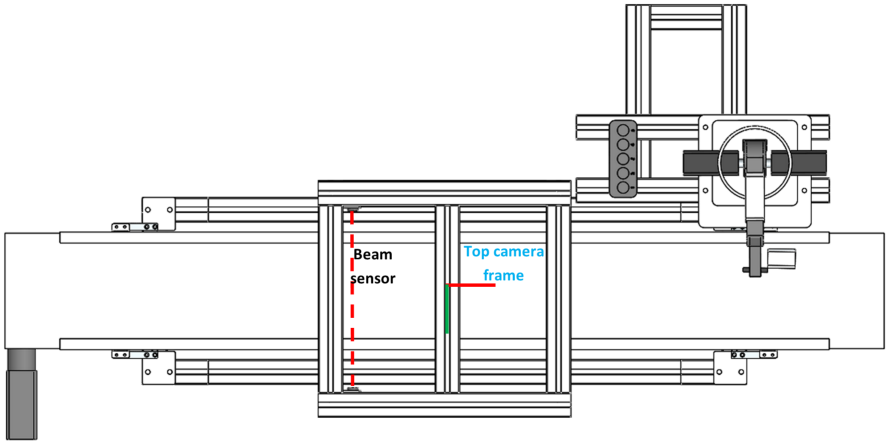


***Figure 9.28*. Illustration of beam sensor position relative to top-view camera frame I_t. Red and green express the x- and y-axes, respectively.**

The maximum operating frequency of the implemented receiver was 500 Hz. As mentioned, the conveyor belt coordinate space is framed in motor steps. The microcontroller tracks the current step location as the conveyor is operated. As the assessed object is stepped toward the vision system, the beam is interrupted. This initial object-beam incidence step location is denoted as N_i. As the object continues past the beam sensor, the beam can resume uninterrupted. This change in signal is transmitted to the microcontroller, which notes the associated step position. The step location at which the beam resumes is denoted as N_s. This process is graphically depicted in Figure 9.29.

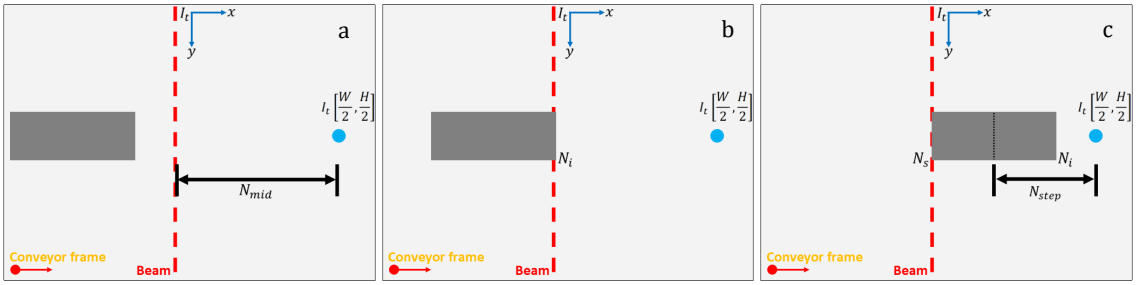


***Figure 9.29*. Illustration of the beam sensor process used to measure object length. (a)—depiction of object heading toward the vision frame. (b)—depiction of initial object-beam incidence. (c)—depiction of beam resuming as object moves past sensor. By recording the step locations of change in beam signal, object length may be estimated.**

By recording the step count at N_i and N_s, the length of the object may be approximated in steps as $N_s - N_i$. The distance from the end of the beam sensor to I_t x-axis centre N_{mid} is known *a priori*. Therefore, the number of steps required to translate the object to $I_t[x] = [960]$ is calculated as:

$$N_{step} = N_{mid} - \frac{N_s - N_i}{2} \tag{9.16}$$

N_{mid} was measured as 157 mm, or 1,640 steps. Object length approximated in the above manner does not necessarily relate to a true dimension of the object. This length simply refers to the number of beam steps interrupted by the object. Consequently, object orientation does not affect the accuracy of this process. The accuracy associated with positioning an object at I_t x-axis centre is also not affected by object orientation. Note that the grasp synthesis methodology proposed in this thesis does not require objects to be

situated in the centre of the vision system, although, the clarity of the camera is ideal at $I_t[x, y] = \left[\frac{W}{2}, \frac{H}{2}\right]$. Therefore, overall measurement error can be reduced by imaging objects at the centre of the vision system.

The error associated with this subsystem was found by comparing an object of known length to the length determined by the beam sensor measurement. It should be observed that this error varied depending on the reflectivity of the object. The beam sensor process consistently translated objects to the centre of $I_t[x]$ within 15 pixels. This value was determined by measuring the distance from $I_t[x] = \left[\frac{W}{2}\right]$ to object centroid $I_t[x_o]$ for circular objects. Measurement errors associated with this subsystem are recorded in Table 9.14.

***Table 9.14*. Error associated with the beam sensor length measurement, taken from 20 measurements.**

Component	Associated error
Beam sensor measurement error (mm)	$\pm$ 1.0 mm
Beam sensor translation accuracy (pixels)	$\pm$ 15.0 pixels

The PA18C beam sensor package is rated with a maximum response time of 1 ms. The nominal operating speed of the conveyor belt is 5,155 steps/s. In 1 ms, the conveyor has moved by approximately 5 steps. Therefore, the maximal possible accuracy of this subsystem is roughly 0.5 mm. It is worth noting that the above measurements are only used to translate the object to I_t x-axis centre. Once the object is imaged by the top-view camera, object position becomes relative to the I_t vision frame as per the grasp synthesis methodology described in Chapter 4.

9.3.6 Grasp-induced error measurement

A significant aspect of this thesis relates to the proposed grasp quality scores, OS and OE, referred to as similarity metrics. This section relates to the implementation details regarding the measurement techniques described in Chapter 3.2. OS and OE are calculated through vision and quantify the displacement caused by the implemented grasp G in terms of overlap and orientation. This displacement is calculated by comparing pre- and post-manipulation images of an object subjected to a physical grasp attempt. The top-down binary image $I_{t_{Th}}$ described in Section 4.3 is used to compute OS and OE. Here, $I_{t_{Th}}$ relates to the object pre-grasp. The post-grasp counterpart is denoted as $I_{t_{Th_{post}}}$. Pre- and post-grasp examples are illustrated in Figure 9.30.

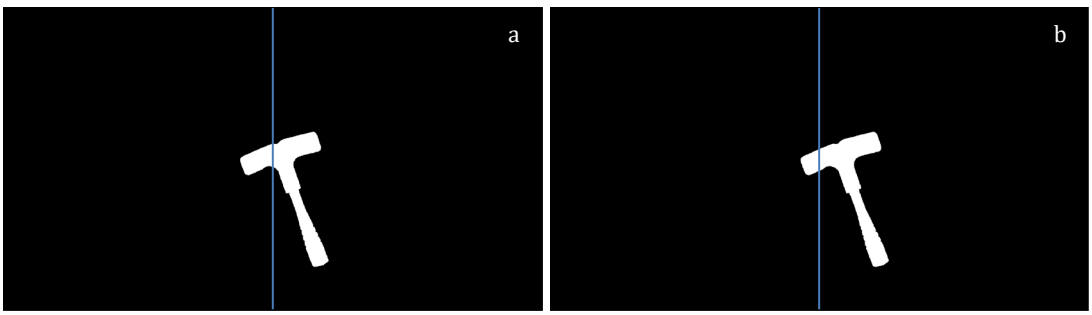


***Figure 9.30*. Example of translational error during prototype implementation. (a)—binary image pre-grasp $I_{t_{Th}}$. (b)—binary image post-grasp $I_{t_{Th_{post}}}$. The vertical line was added to provide a point of reference.**

Object region areas within $I_{t_{Th}}$ and $I_{t_{Th_{post}}}$ are computed as:

$$A_{pre} = \sum_{x=1}^{W} \sum_{y=1}^{H} I_{t_{Th}}[x, y], \quad if\ I_{t_{Th}}[x, y] = 1 \tag{9.17}$$

$$A_{post} = \sum_{x=1}^{W} \sum_{y=1}^{H} I_{t_{Th_{post}}}[x, y], \quad if\ I_{t_{Th_{post}}}[x, y] = 1 \tag{9.18}$$

A_{pre} and A_{post} relate to the number of pixels within the respective binary image that register a value of 1. Generally, A_{pre} and A_{post} register approximately the same value, as assessed objects do not change in size between examinations. To quantify the overlap between $I_{t_{Th}}$ and $I_{t_{Th_{post}}}$, a new binary image is created by assessing each pixel at shared coordinates. If both images register a value of 1 at the same location, the new image coordinate value is set to 1:

$$I_{t_{Th_{int}}}[x, y] = 1, \qquad if \begin{cases} I_{t_{Th}}[x, y] = 1 \\ I_{t_{Th_{post}}}[x, y] = 1 \end{cases} \tag{9.19}$$

The intersection over union between pre- and post-grasp images may be computed by counting the number of pixels of value 1 within binary intersection image $I_{t_{Th_{int}}}$:

$$A_{pre} \cap A_{post} = \sum_{x=1}^{W} \sum_{y=1}^{H} I_{t_{Th_{int}}}[x, y], \quad if\ I_{t_{Th_{int}}}[x, y] = 1 \tag{9.20}$$

$A_{pre} \cap A_{post}$ relates to the pixel area of the region within $I_{t_{Th_{int}}}$, illustrated in Figure 9.31—b. For clarity, an absolute difference image between pre- and post-grasp images is shown in Figure 9.31—a.

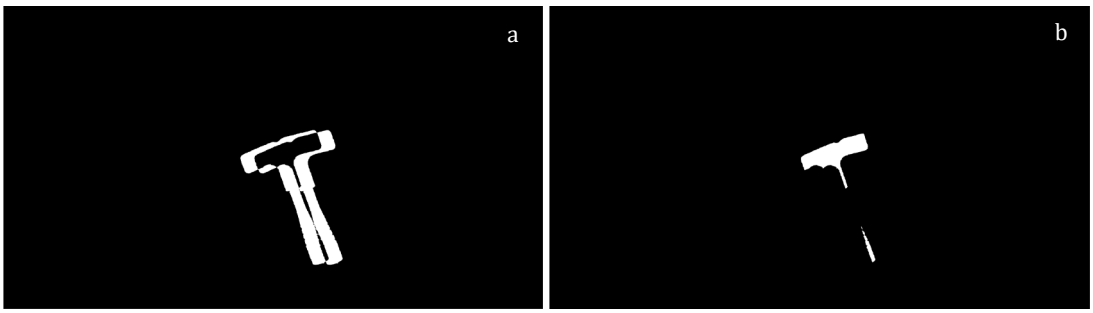


Figure 9.31. **(a)—absolute difference image comparing pre- and post-grasp images of an object. (b)—intersection image $I_{t_{Th_{int}}}$. This image highlights the intersection over union between pre- and post-grasp images.**

Thus, the values needed to calculate the *OS* score in Equation 3.1, Section 3.2 are satisfied. The orientation of an object in binary space is computed as the angle between the I_t x-axis and the major axis of an ellipse with the same second-order moments as the object region, graphically depicted in Figure 9.32.

***Figure 9.32*. Example of orientational error during prototype implementation. (a)—pre-grasp image. (b)—post-grasp image. Red represents the fitted ellipse containing the object. Blue represents ellipse axes. Green represents ellipse angle with respect to I_t x-axis.**

Deriving pre-grasp angle θ_{pre} and post-grasp angle θ_{post} is mathematically complex. A description of this derivation is beyond the scope of this thesis. For a comprehensive report covering the moment analysis of binary regions, refer to J. Kilian [415]. Computing the major orientation of a binary region is a common task within digital image processing. Consequently, the standard Matlab *regionprops* function computes object orientation, among several other properties related to image regions. Therefore, θ_{pre} and θ_{pre} are easily derived, satisfying the OE score calculation described by Equation 3.2, Section 3.2. Figures 9.33 and 9.34 illustrate some observed examples related to the response of OS and OE, respectively.

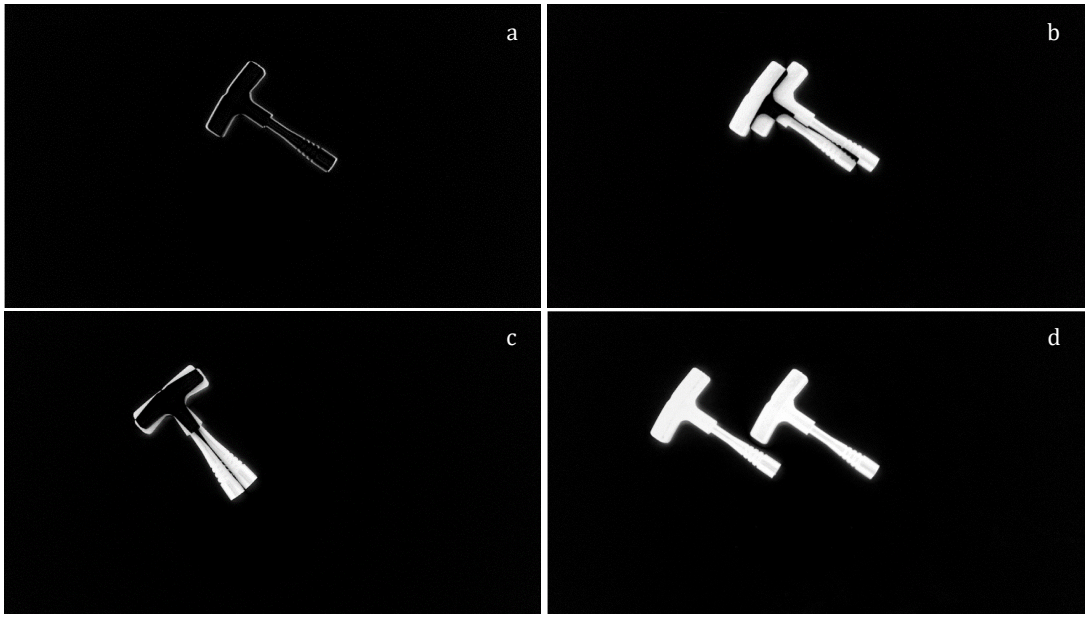
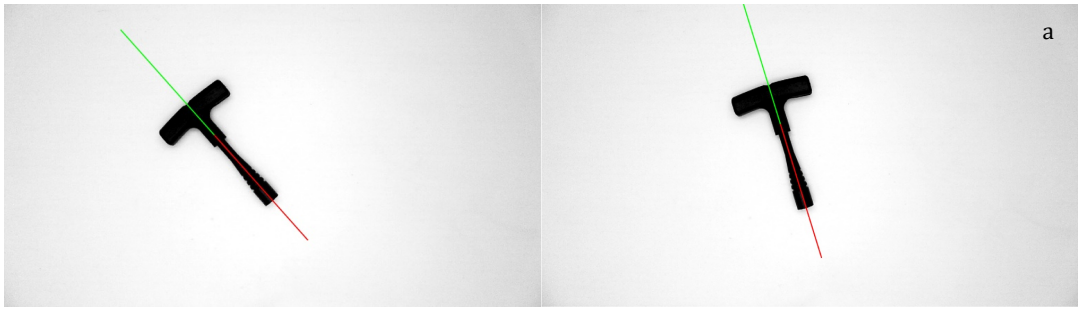


***Figure 9.33*. Examples of pre- and post-grasp images overlaid. (a)—high-scoring OS pair of images. (b)—low-scoring OS pair of images. (c)—OS score affected by orientational change. (d)—no overlap, therefore OS score of 0.**



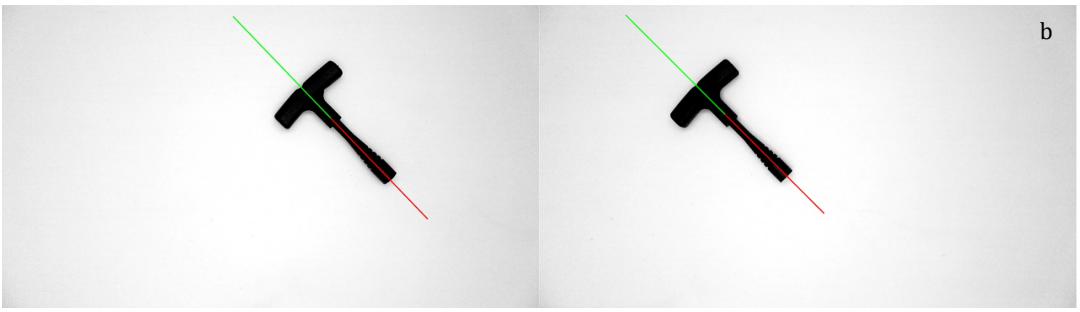


***Figure 9.34*. Examples of orientational error measurements between pre- and post-grasp images. (a)—significant orientation error introduced by grasp attempt, resulting in low OE. (b)—example of no orientational error introduced by grasp attempt. Note OE does not respond to changes in translation.**

Notably, the OE score does not respond to translational change between pre- and post-grasp images. The example shown in Figure 9.34—b for instance, suffers no loss in OE because of the introduced translational error. It was expected that OE scoring would not be able to reliably quantify the rotational offset between images for perfectly circular objects. This was not an issue during implementation however, as the precision of the vision system detected slight nuances in objects perceived as perfectly circular by the human operator, e.g., kH001 (Sellotape adhesive tape), uC004 (large rubber seal), uC011 (circular black servo component) and mC014 (ball bearing).

Due to noise present in the vision system, conveyor and robotic manipulator, OS and OE assessments were prone to minor errors in measurement. Table 9.15 lists the amount of error associated with each respective component.

***Table 9.15*. Error associated with OS and OE, found from 20 measurements per type of error for a total of 120 samples.**

	Vision	Vision + conveyor	Vision + conveyor + robot
OS	$\pm$ 0.0012	$\pm$ 0.0204	$\pm$ 0.0545
OE	$\pm$ 0.0001	$\pm$ 0.0007	$\pm$ 0.0033

Error associated with the vision system was measured by computing OS and OE for stationary objects subjected to no translation or manipulation action. Conveyor error was measured by translating objects to robot $\bar{G}$ x-axis zero, then back to the original position in image space I_t. Finally, robot error was measured through a pick-and-place action for an object pre-placed by the manipulator, thus avoiding the measurement of error that may be introduced by an incorrect grasping location.

9.4 Experimental protocol

To trial the proposed 3-stage grasp synthesis methodology deployed on the discussed apparatus, single objects were placed haphazardly on the conveyor belt furthest from the robotic manipulator. Detected objects were translated to I_t x-axis centre utilising the beam sensor process. Once stationary, objects were analysed by vision and the load-cell subsystem to generate many potential grasping locations. After finalising a grasp pose via the employed selection criterion, the object was translated into robot space, where it was subjected to a manipulation procedure. Conveyor translation steps are detailed in Section 9.3.1. Depending on the outcome, the trial may be labelled as either $pass$ or $fail$. Post-manipulation, the object is translated back into vision space to assess the quality of the performed grasp in terms of OS and OE. The above described process constitutes one trial.

In the manipulation procedure, autonomously computed grasp poses $G[x_t, y_t, y_s, \theta_t]$ were physically attempted by the robotic manipulator. The object is grasped and lifted vertically to a height of 15 cm. The object is then suspended for 10 seconds, after which it is placed back on the conveyor platform at the initial grasp location $\bar{G}[\bar{x}, \bar{y}, \bar{z}, \bar{\theta}]$. For a grasp trial to be labelled as $pass$ or 1, the object must not fall at any stage of the trial. Moreover, objects must be solely supported by the gripper, such that no other part of the object touches any other hardware. Any trial which does not conform to the outlined criteria is labelled as $fail$, or 0. This definition of a $pass$ label is derived from the usage throughout literature, described in Chapter 2.6 and further discussed in Chapter 10.3. This research intends to quantify grasp outcome by using the proposed OS and OE measurements, moving away from constrained binary measures such as grasp success rates. These scores are continuous, thus providing a spectrum related to the error introduced by the grasp process. This distinction allows further quality assessment for grasps that are already labelled as $pass$. Chapter 3 discusses this topic in detail. For comparison purposes and as an additional metric to assess performance, the $pass/fail$ definition of grasp outcomes is recorded.

The 3-stage grasp synthesis methodology was trained using known objects exclusively. However, testing utilised the entire 100-object test pool listed in Chapter 5—comprised of known and unknown objects. An object is considered known if a sample related to that object appears in any training data. 15 objects were used to create training data. The remaining 85 were used for testing purposes only. This division sets a baseline to gauge the generalised handling performance of the system when presented with novel objects.

Depending on the type of trial, the grasp to be implemented $G[x_t, y_t, y_s, \theta_t]$ is selected in various ways from candidate grasp matrix g produced by the stage 2 framework. The grasp generation and selection methodology is discussed in Chapter 4.3. To measure the overall performance of the proposed approach—and individual aspects—four separate selection criteria were trialled. Table 9.16 details the various selection criteria employed in each respective round of testing.

***Table 9.16*. Description of the four types of selection criteria used during trials.**

Selection criterion	Description
Random	The grasp to be implemented G is randomly selected from candidate grasp matrix g.
Highest $K_{I_{RGW}}$	The highest $K_{I_{RGW}}$ scoring sample from candidate grasp matrix g is selected for implementation. $K_{I_{RGW}}$ denotes the softmax function output of the stage 2 CNN for the *positiveGrasp* class, recorded in score matrix S_c.
Highest $pass_{pred}$	Implemented grasps G are selected from grasp matrix g by a network trained to predict the probability of a $pass$ label based on the 10 input scores from S_c. The candidate with the highest probability of resulting in a $pass$ label is selected for implementation.
Highest combined OS_{pred}, OE_{pred} (Stage 3 selection)	A grasp G is selected from matrix g by the stage 3 framework, which predicts OS and OE scores given candidate sample score matrix S_c. The candidate with the highest combined predicted scores, OS_{pred} and OE_{pred}, is selected for implementation.

The random selection criterion was chosen to define a baseline that represents a methodology in which minimal selection pressure is applied. The resulting outcome of

physical trials with this criterion relates to the average grasp recognition rate of the stage 2 classifier and relevance of the associated training set. Selection via highest $K_{I_{RGW}}$ represents a methodology in which samples are selected by the stage 2 framework, i.e., the sample most strongly associated with the user-defined *positiveGrasp* class is selected. The *positiveGrasp* class is intended to reflect grasp areas that result in a $pass$ label when implemented. Therefore, this selection procedure aims to maximise for grasp rate with input I_{RCW} based on human expertise. Many works throughout literature employ this selection strategy. In addition to the above-mentioned selection criteria, several networks were trained to predict the outcome of a given sample using the data from the stage 3 training set. Selection via $pass_{pred}$ utilises a network that predicts the probability of a $pass$ label outcome. Conducting trials with the highest $pass_{pred}$ standard aims to gauge the performance of a methodology trained to optimise for grasp rate with input S_c. Highest combined OS_{pred}, OE_{pred} selection maximises for the grasp quality measurements, OS and OE, and aims to evaluate the performance of the proposed selection process.

A preliminary round of testing was conducted in the earlier stages of development. In this round of trials, the object test pool was constituted of 10 known objects and 18 unknown objects. The object test pool register utilised in the initial round of testing is listed in Table 9.17.

***Table 9.17*. Tentative object register for the first round of testing (R1).**

Series	ID	Description
Known	kH005	Pencil
	kT001	Electrical switch
	kT003	Large hex key
	kT004	Small adjustable wrench
	kT005	Small side cutters
	kC001	Small silver bolt
	kC002	Rod end bearing
	kC003	Pull start handle
	kC004	Pneumatic T-splitter
	kC005	XCPC pneumatic valve
Unknown	uH010	Cruze glide USB
	uH005	Bic permanent marker
	uC004	Large rubber seal
	mC012	Green terminal black
	mC009	Black pneumatic line
	uT005	Blue and yellow screwdriver
	uC003	Small black component
	-	Black acrylic half-circle
	uC013	Silver eyelet screw
	-	Brass pneumatic nozzle
	mT005	Blue Officemax pen
	mT003	Wooden handle
	uC002	Aluminium corner component
	kT002	Small cross wrench
	uC009	Large blue nylock nut
	uC015	VGA connector
	uT008	Small pliers
	uC006	Large pneumatic nut

A total of 940 trials were conducted using the four selection criteria described in Table 9.16. 100 grasps were implemented via random selection. 10 grasps were attempted for the remaining selection criteria for each of the 28 objects within the tentative object pool, for a total of 280 grasp attempts per selection method. The number of trials per selection criteria for the first round of testing is shown in Table 9.18.

***Table 9.18*. Number of trials per selection criterion for first round of testing (R1).**

Selection criterion	Trials
Random	100
Highest $K_{I_{RGW}}$	280
Highest $pass_{pred}$	280
Highest combined OS_{pred}, OE_{pred} (Stage 3 selection)	280

Networks employed in the first round of testing were trained using earlier datasets that contained samples only relating to the 10 known objects within the tentative object register. Stage 1, stage 2 and stage 3 learning components were trained using the STG1_v3 (Table 6.3, Chapter 6.1), STG2_v3 (Table 7.2, Chapter 7.1) and STG3_v1 (Table 8.1, Chapter 8.1) datasets, respectively. Network usage and training details related to the initial round of testing are available in Table 9.19.

***Table 9.19*. Implemented networks and datasets for the preliminary round of testing (R1).**

Stage	Component	Details
Stage 1	Network	*97classNet_V3_11*
	Validation accuracy	97.00%
	Dataset	STG1_v3
	Sample size	11,000
Stage 2	Network	*graspNet_V3_10*
	Validation accuracy	98.01%
	Dataset	STG2_v3
	Sample size	33,000
Stage 3	Network	*OSpredNet_V1*
	RMSE	0.17
	Network	*OEpredNet_V1*
	RMSE	0.14
	Dataset	STG3_v1
	Sample size	500
	Network	*passpredNet_V1*
	RMSE	0.12
	Dataset	STG3_v1
	Sample size	500

The candidate grasp matrix g was populated by the *graspNet_V3_10* CNN for the first round of testing. The softmax output of this CNN was used for highest $K_{I_{RGW}}$ selection. The *passpredNet_V1* network was used for highest $pass_{pred}$ selection. Regression models, *OSpredNet_V1* and *OEpredNet_V1*, were employed for highest combined OS_{pred}, OE_{pred} selection.

A second round of trials was conducted with improved training datasets, retrained networks and the entire 100-object test pool. Each selection criterion was trialled 10 times

per object, for a total of 4,000 physically attempted grasps. The number of trials per selection criterion is listed in Table 9.20.

***Table 9.20*. Number of trials per selection criterion for second round of testing (R2).**

Selection criterion	Trials
Random	1,000
Highest $K_{I_{RGW}}$	1,000
Highest $pass_{pred}$	1,000
Highest combined OS_{pred}, OE_{pred} (Stage 3 selection)	1,000

The *graspNet_V4_15* CNN was utilised in this round of testing to populate the grasp matrix g, trained using the STG2_v4 dataset (Table 7.1, Chapter 7.1). The softmax output of this network was used for highest $K_{I_{RGW}}$ selection. *passpredNet_V3* was employed for highest $pass_{pred}$ selection. *OSpredNet_V3* and *OEpredNet_V3* were used to facilitate stage 3 selection. The dataset used to train the selection stage networks is detailed in Table 8.2, Section 8.1. Network usage and training details related to the second round of testing are listed in Table 9.21.

***Table 9.21*. Implemented networks and datasets for the second round of testing (R2).**

Stage	Component	Details
Stage 1	Network	*classNet_V4_16*
	Validation accuracy	99.51%
	Dataset	STG1_v4
	Sample size	80,000
Stage 2	Network	*graspNet_V4_15*
	Validation accuracy	99.41%
	Dataset	STG2_v4
	Sample size	141,000
Stage 3	Network	*OSpredNet_V3*
	RMSE	0.07
	Network	*OEpredNet_V3*
	RMSE	0.14
	Dataset	STG3_v3
	Sample size	2,000
	Network	*passpredNet_V3*
	RMSE	0.08
	Dataset	STG3_v3
	Sample size	2,000

For quantitative results pertaining to trial round 1 and trial round 2, refer to Section 10.1. Section 10.2 offers a qualitative analysis.

Chapter 10

Results and discussion

10.1 Quantitative analysis

The first round of testing utilised the methodology, prototype and experimental protocol outlined in this research. However, earlier networks trained on smaller datasets were employed in the learning components of stage 1, stage 2 and stage 3. Moreover, a reduced 28-object test pool was used in the initial round of testing. Four separate selection criteria were evaluated. For more information regarding the experimental procedure, refer to Section 9.4. Implementing a random grasp G from grasp candidate matrix g resulted in a successful trial 95.0% of the time, for known and unknown objects. This aggregate grasp rate was 96.8% when selecting a grasp candidate based on highest softmax output $K_{I_{RGW}}$. Selection via highest $pass_{pred}$ yielded an aggregate grasp rate of 93.7%. Finally, the recorded grasp rate was 99.3% when the proposed stage 3 selection process was implemented, i.e. highest combined OS_{pred}, OE_{pred}. Grasp rate refers to the percentage of trials labelled as $pass$. The outcome of the first trial by $pass/fail$ label is graphed in Figure 10.1.

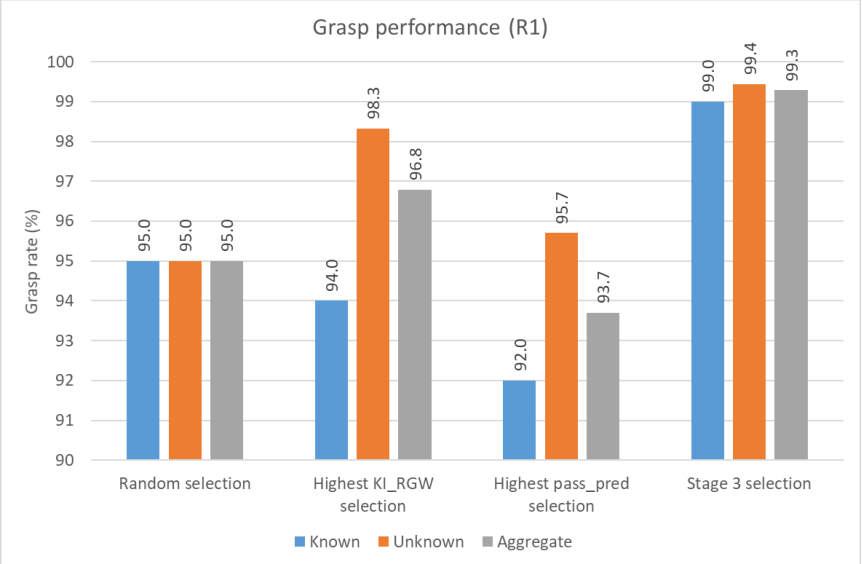


Figure 10.1. **Grasp rate from the first round of testing by selection criterion and object series (940 trials).**

Prior to trials, it was hypothesised that the highest $pass_{pred}$ criterion would result in similar or even improved grasp rates than highest $K_{I_{RGW}}$ selection. This was expected since the network employed in $pass_{pred}$ selection utilises $K_{I_{RGW}}$ in addition to the nine other scores in S_c—thus, the $pass_{pred}$ network is privy to more information when predicting grasp outcome. At a minimum, this mode of selection was expected to perform at the rate of random selection as this method does not exhibit any selection action between samples. Due to the tentative nature of the first trial, only 100 grasps were attempted via random selection and 280 for the other criteria. The true grasp rate when randomly selecting from

candidate grasp matrix g, generated using *graspNet_V3_10* may be lower or equal to the recorded grasp rate and more testing may be needed to explain this discrepancy.

For selection via highest $K_{I_{RGW}}$, a 4.3% improvement in grasp rate was recorded for the 18 unknown objects, compared to grasping from the 10 known object subset. This was not expected as the network used to generate candidate grasps *graspNet_V3_10* was trained with samples from the known set exclusively. Therefore, objects within the unknown set are grasped using features learned from the known set, i.e. *no a priori* knowledge is available for objects within the unknown set. Therefore, it was anticipated that the grasp rate would be higher for objects from which the training data was derived. However, the features learned when utilising the STG3_v1 dataset are not object-dependent and aimed to maximise for *positiveGrasp* or *negativeGrasp* label classification accuracy. Learning to detect grasps is discussed in Chapter 7.2. Highest $pass_{pred}$ selection also achieved better grasp rates for objects within the unknown set. The higher grasp rate observed for objects seen for the first time may be tied to the manipulation difficulty of the objects within each respective subset. To minimise this gap, a larger variety of objects may need to be tested. Note that this difference in grasp rate was mitigated by stage 3 selection.

Compared to the random selection baseline, highest $K_{I_{RGW}}$ improved performance by approximately 2%. This performance relates to the classification accuracy of *graspNet_V3_10*. Grasping via $pass_{pred}$ selection reduced grasp rates by approximately 1%. The overall grasp rate was improved by approximately 4% when implementing a framework trained to maximise for OS_{pred} and OE_{pred}. Though the utility of selecting for OS and OE is reflected by the grasp rate, grasp quality is more accurately described by the proposed metrics. Figure 10.2 graphs some examples of typical OS and OE scores for various objects deemed as successfully grasped by the $pass/fail$ metric.

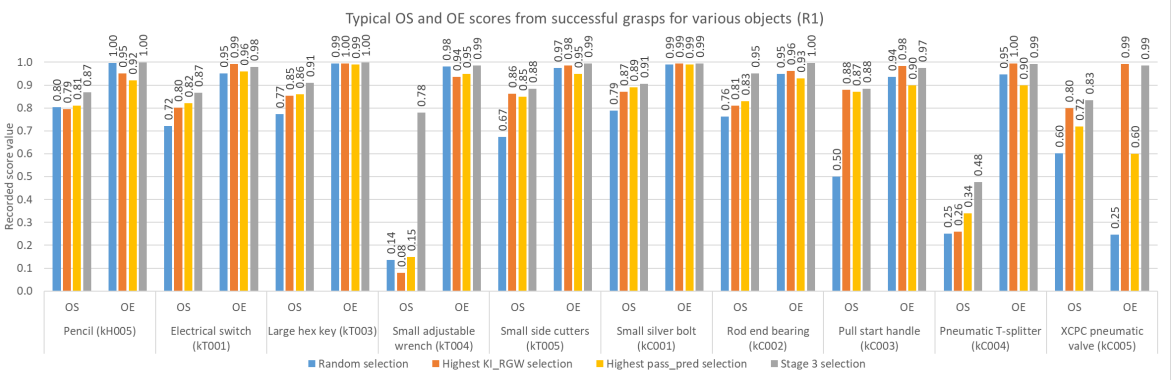


***Figure 10.2*. Typical OS and OE scores for objects labelled as $pass$ during the first round of trials by selection criterion.**

Generally, stage 3 selection resulted in higher OS and OE scores. This was expected since this mode of selection is trained to optimise for these scores. This improvement was particularly relevant in the case of the small adjustable wrench (kT004). Random selection, highest $K_{I_{RGW}}$ selection and $pass_{pred}$ selection generally resulted in very poor OS scores for this object. Note that the scores shown in Figure 10.2 relate to grasps labelled as $pass$ exclusively. Due to the uneven mass distribution of the wrench, it became increasingly important to grasp the object close to the COG. Highest $K_{I_{RGW}}$ selection tended toward the handle, resulting in significant droop and low-quality grasps. Though such grasps were qualitatively noticeable as poor, the $pass/fail$ metric could not quantify this disparity.

Grasps selected via highest combined OS_{pred}, OE_{pred} tended toward the COG of this object, significantly improving the resultant OS score. Conversely, grasp location was significantly less important for the pencil object (kH005). Because of the low mass and parallel grasping surface, grasps for this object tended to be very robust, regardless of location. The pneumatic flow valve (kC005) was a notably difficult object to grasp successfully during the first round of trials due to false positives produced by *graspNet_V3_10*. Though such grasps were often selected for implementation via the random, highest $K_{I_{RGW}}$ and highest $pass_{pred}$ criteria, they were largely avoided by stage 3 selection. Generally, grasps selected via highest $pass_{pred}$ yielded higher resultant OS scores, compared to highest $K_{I_{RGW}}$. However, the $K_{I_{RGW}}$ criterion tended to be associated with higher resultant OE.

Analysis of OS and OE for the first round of trials is based on the input score matrix S_c and resultant output scores OS, OE and $pass/fail$ recorded in the STG3_v1 dataset. Samples within this dataset were collected by randomly implementing grasps from candidate matrix g and recording the resultant outcome. Therefore, this dataset is analogous to a trial in which random selection was employed for known objects. The distributions of output scores OS and OE by $pass/fail$ label for the STG3_v1 dataset are illustrated in Figure 10.3.

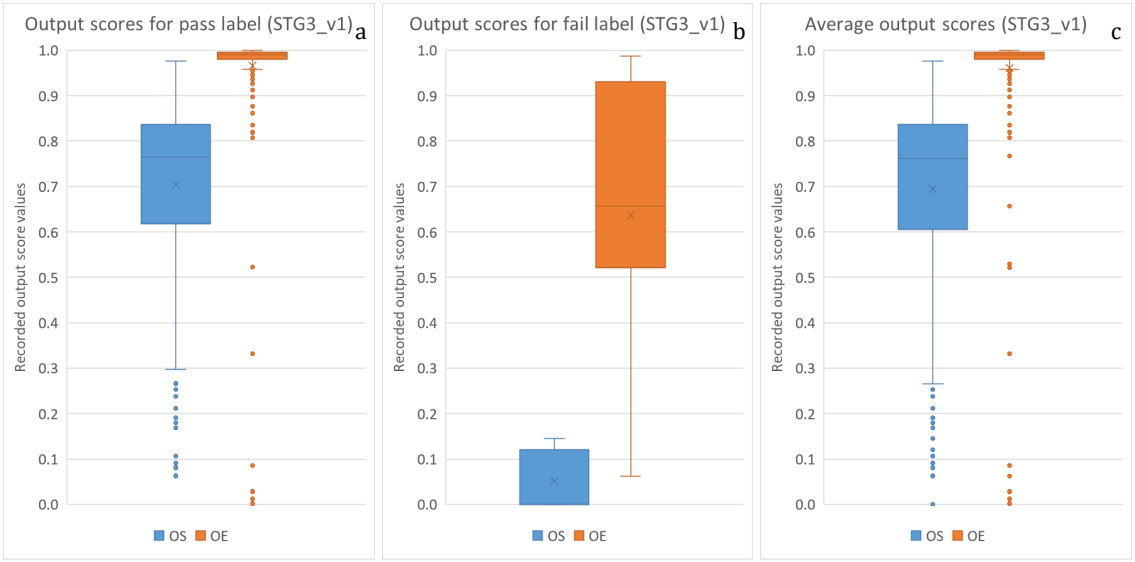


***Figure 10.3*. Distribution of recorded output scores OS and OE by $pass/fail$ label from the STG3_v1 dataset (500 samples). (a)—output score distributions for the $pass$ label. (b)—output score distributions for the $fail$ label. (c)—average output score distributions recorded in the STG3_v1 dataset by label.**

Figure 10.3—a illustrates the wide range of resultant overlap scores OS for the $pass$ label. The orientation score OE, in contrast, tended to result in scores above 0.95 for grasps considered successful by the $pass/fail$ definition. However, trials labelled as $fail$ were generally associated with low OS scores, but no specific range in OE score (Figure 10.3—b). Overall, OS and OE scores recorded in the STG3_v1 dataset varied significantly. This dissertation is particularly interested in the spread of OS and OE for samples considered successful and argues that the $pass/fail$ measure is only one metric for consideredation. Though a grasp configuration may be selected such that the resulting trial is classified as $pass$, it is equally pragmatic to select grasps such that the resultant manipulation and placement accuracy is not compromised, quantified by OS and OE. As discussed in Chapter 2, grasp rate alone cannot be used to assess grasp quality, which is needed in more sophisticated grasping applications throughout industry.

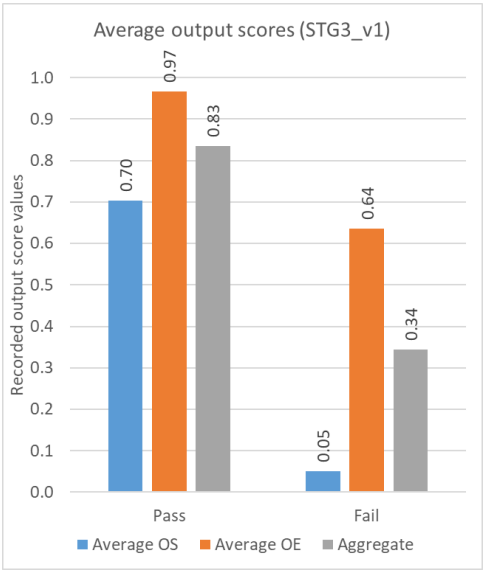


***Figure 10.4*. Average output scores by $pass/fail$ label from the STG3_v1 dataset (500 samples).**

The graph shown in Figure 10.4 relays the average OS and OE scores from random selection for known objects, by $pass/fail$ label. Generally, grasps labelled as $pass$ scored significantly higher OS and OE than the $fail$ label. The average OS and OE scores recorded in the STG3_v1 dataset for a failed grasp were 0.05 and 0.64, respectively. The average OS and OE scores recorded in the STG3_v1 dataset for a successful grasp were 0.70 and 0.97, respectively. Figure 10.4 shows that very little orientational error was introduced for grasps labelled as $pass$. Grasps labelled as $fail$ generally resulted in very low OS scores. This suggests a relationship between the $pass/fail$ literature definition of grasp outcome and the proposed similarity metrics, OS and OE.

To investigate the utility of the proposed selection stage networks *OSpredNet_V1* and *OEpredNet_V1*, OS_{pred} and OE_{pred} were predicted from the input candidate sample score matrix S_c recorded for each sample in the STG3_v1 dataset. The distributions of predicted output values for samples from the STG3_v1 dataset are shown in Figure 10.5.

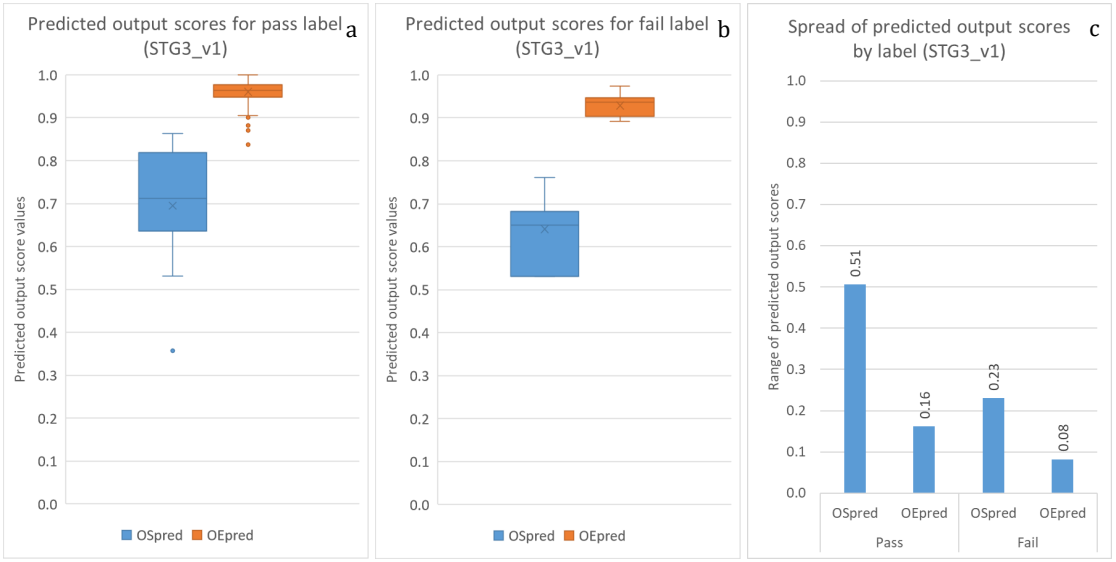


Figure 10.5. **Distribution of predicted output scores OS_{pred} and OE_{pred} by $pass/fail$ label given input scores from the STG3_v1 dataset. (a)—predicted output score distributions for the $pass$ label. (b)—predicted output score distributions for the $fail$ label. (c)—spread of predicted output scores by label.**

The spread of predicted output scores OS_{pred} and OE_{pred} for samples within the $pass$ label were 0.51 and 0.16, respectively. For samples labelled as $fail$, the spread of OS_{pred} and OE_{pred} were 0.23 and 0.08, respectively. Generally, predicted scores were not as polarised as their measured counterparts recorded in the STG3_v1 dataset. The stage 2 component utilised in this trial tended to generate approximately 68 candidate grasps per object. To appropriately select a sample for implementation from this pool, it is essential to employ predictors that can sufficiently differentiate between samples. The range of predicted output scores relates to the sensitivity of the utilised networks, provided network accuracy is acceptable. This quality discrimination allows for the optimisation of OS and OE, and by extension, grasp rate, quantified by the $pass/fail$ metric. Note that the capacity of OS and OE to predict a $pass/fail$ label is irrelevant to this work. It is not the intention to define a relationship between OS, OE and the $pass/fail$ metric, but rather to find relationships between input features related to the object in the form of score matrix S_c and the proposed similarity scores, OS and OE. Framing grasp attempts as successful or unsuccessful is useful for literature comparison and serves as an additional metric to illustrate the utility of OS and OE.

In the first round of experimentation, 940 grasp attempt trials were conducted for four separate selection criteria. While this number is high compared to other methodologies throughout literature, a relatively low degree of confidence was associated with some of the grasp rate estimates for this sample survey. At 95% confidence, the margin of errors are ± 4.27%, ± 2.06%, ± 2.85% and ± 0.98% for grasping via random selection, highest $K_{I_{RGW}}$ selection, highest $pass_{pred}$ selection and highest combined OS_{pred}, OE_{pred} selection, respectively.

To address some of the concerns noted in the first round of trials, a second round was conducted. Round 2 consisted of 4,000 trials of the 100-object test pool documented in Chapter 5. Each object was trialled 10 times per selection criteria, resulting in 1,000 trials for random selection, 1,000 trials for selection via highest $K_{I_{RGW}}$, 1,000 trials for highest $pass_{pred}$ selection and 1,000 trials for the highest combined OS_{pred}, OE_{pred} mode of

selection. Grasping known and unknown objects via random selection resulted in a $pass$ outcome 94.0% of the time. Highest $K_{I_{RGW}}$ improved this grasp rate to 96.4%. Selecting via highest $pass_{pred}$ achieved a similar aggregate outcome of 96.3%. The stage 3 selection criterion resulted in an aggregate grasp rate of 99.0%. A histogram illustrating the results of the second trial in terms of grasp rate is shown in Figure 10.6.

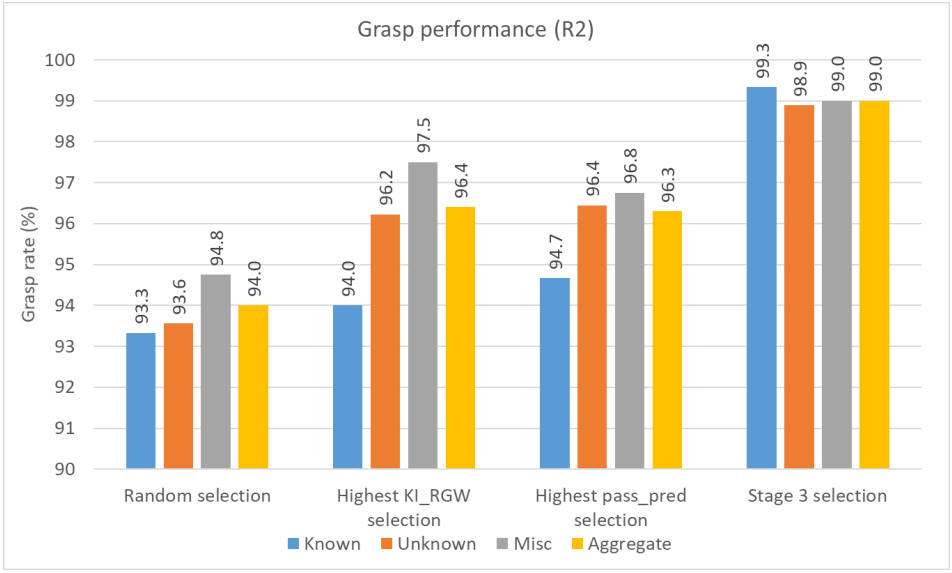


***Figure 10.6*. Grasp rate from the second round of testing by selection criterion and object series (4,000 trials).**

Grasping via random selection yielded a 93.3% grasp rate for the known object series, increasing slightly to 93.6% and 94.8% for the unknown and misc. subsets, respectively. Highest $K_{I_{RGW}}$ selection followed a similar pattern, with a resultant grasp rate of 94.0% for known objects. Significantly improved grasp rates were recorded for this mode of selection for the unknown and misc. series, resulting in 96.2% and 97.5%, respectively. Highest $pass_{pred}$ selection shared a similar trend, with recorded grasp rates of 94.7%, 96.4% and 96.8% for the known, unknown and the misc. series, respectively. Note that the relative increase in grasp rate by object series was consistent across selection criteria for random, $K_{I_{RGW}}$ and $pass_{pred}$ selection. This pattern may be related to the difficulty of each respective subset. Selection via stage 3 resulted in the least amount of variation across object sets, with recorded performances of 99.3%, 98.9% and 99.0% for the known, unknown and the misc. series, respectively.

At 95% confidence, the margin of error for the aggregate result of grasping via random selection was ± 1.47%. For highest $K_{I_{RGW}}$ selection, highest $pass_{pred}$ selection and highest combined OS_{pred}, OE_{pred} selection, this margin was reduced to ± 1.15%, ± 1.17% and ± 0.62%, respectively. Compared to the initial round of trials, a higher degree of confidence is associated with the results from the sample survey conducted in the second round. Despite the difference in object pool, the results from each respective trial follow a relatively consistent pattern. The outcome of the second round of trials by series and category is tabulated in Tables 10.1, 10.2 and 10.3.

***Table 10.1*. Outcome of grasp trial round 2 for the known object series in terms of $pass/fail$ label by category.**

Known							
Selection criterion	Household		Tool		Component		Total
	Pass	Fail	Pass	Fail	Pass	Fail	
Random selection	47	3	47	3	46	4	150
Highest $K_{I_{RGW}}$	47	3	47	3	47	3	150
Highest $pass_{pred}$	46	4	47	3	49	1	150
Highest combined OS_{pred}, OE_{pred}	50	0	50	0	49	1	150
							600

The figurine object (kH002) was a notably difficult object to grasp within the known household subset. Random selection failed to grasp this object 3 times, while highest $K_{I_{RGW}}$ resulted in 2 failed grasp attempts. Selection via highest $pass_{pred}$ only failed to grasp this object once. Highest combined OS_{pred}, OE_{pred} however, successfully grasped all objects within the known household subset. The XCPC pneumatic value (kC005) was another difficult object, designated within the known, component subset. Random selection resulted in 4 failures and highest $K_{I_{RGW}}$ selection, 2 failures. This reduction in performance stemmed from the stage 2 network employed in this round of trials. *graspNet_V4_15* was prone to generating false positives for this object, which were generally avoided by highest $pass_{pred}$ selection and stage 3 selection. Graphical examples of grasps selected by each respective criterion are illustrated in Chapter 10.2.

***Table 10.2*. Outcome of grasp trial round 2 for the unknown object series in terms of $pass/fail$ label by category.**

Unknown							
Selection criterion	Household		Tool		Component		Total
	Pass	Fail	Pass	Fail	Pass	Fail	
Random selection	143	7	138	12	140	10	450
Highest $K_{I_{RGW}}$	141	9	142	8	150	0	450
Highest $pass_{pred}$	144	6	144	6	146	4	450
Highest combined OS_{pred}, OE_{pred}	150	0	145	5	150	0	450
							1,800

12 failed grasp attempts were associated with the key object (uH006) for this trial. Random selection resulted in 4 failures, highest $K_{I_{RGW}}$ resulted in 5 failures and highest $pass_{pred}$, 3 failed attempts. All selection strategies struggled with the silver combination lock (uT015). 2 $fail$ labels were recorded for this object for the random selection criterion. 4 $fail$ labels were recorded for highest $K_{I_{RGW}}$. Highest $pass_{pred}$ and stage 3 selection failed to grasp this object 3 and 4 times out of the 10 trials dedicated to this object, respectively.

***Table 10.3*. Outcome of grasp trial round 2 for the misc. object series in terms of $pass/fail$ label by category.**

Misc.							
Selection criterion	Household		Tool		Component		Total
	Pass	Fail	Pass	Fail	Pass	Fail	
Random selection	110	10	97	3	172	8	400
Highest $K_{I_{RGW}}$	113	7	97	3	180	0	400
Highest $pass_{pred}$	111	9	100	0	176	4	400
Highest combined OS_{pred}, OE_{pred}	117	3	100	0	179	1	400
							1,600

The PK chewing gum (mH010) was a household object of notable difficulty within the misc. object series. Grasp attempts resulting in failure for this object were generally related to false positives generated by *graspNet_V4_15*. The $fail$ labels associated with this object for random selection, highest $K_{I_{RGW}}$, highest $pass_{pred}$ and highest combined OS_{pred}, OE_{pred}, were 4, 5, 5 and 2 respectively. A histogram collating the results of the above tables is illustrated in Figure 10.7.

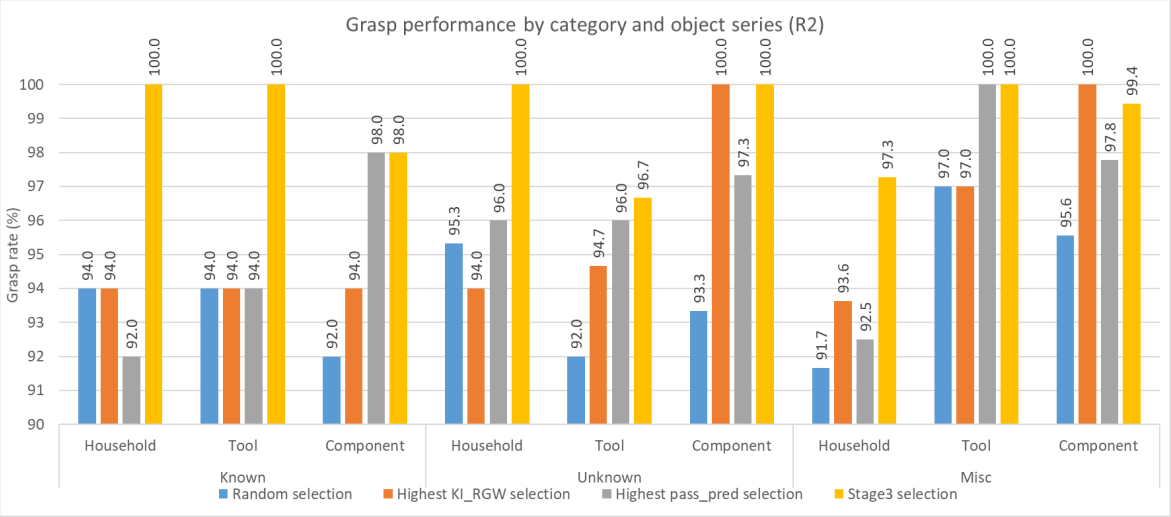


***Figure 10.7*. Histogram related to the grasp rate recorded in the second round of trials by category, object series and selection criterion.**

For a full list of results pertaining to the second round of trials, see Appendix C. To gauge the sensitivity of the stage 3 networks, *OSpredNet_V3* and *OEpredNet_V3*, predicted values OS_{pred} and OE_{pred} were computed for each grasp trial, given the score matrix S_c recorded during the second round of trials. The results are illustrated in Figure 10.8.

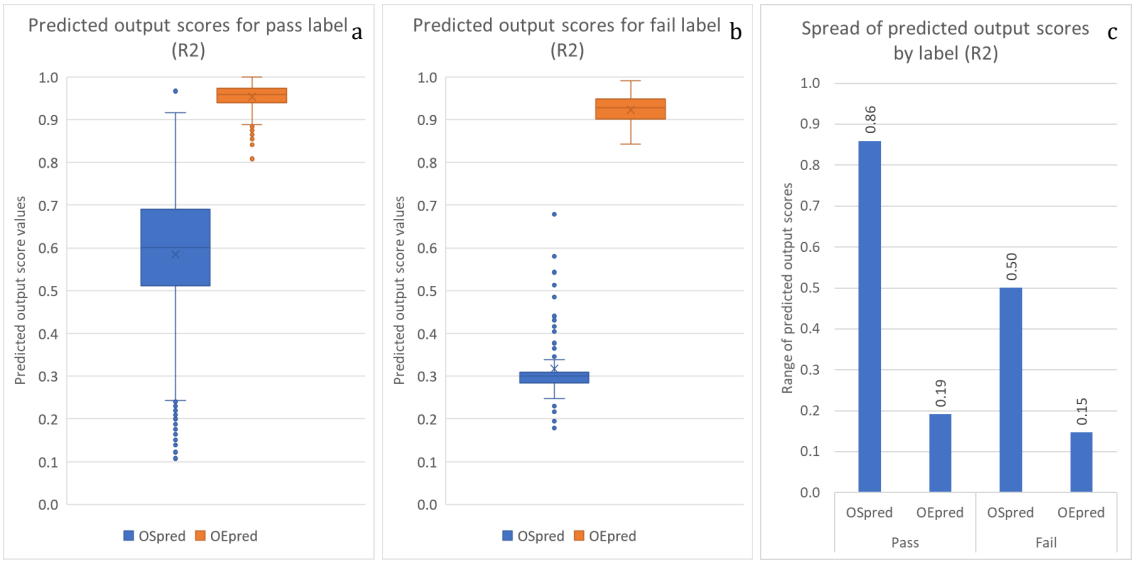


***Figure 10.8*. Distribution of predicted output scores OS_{pred} and OE_{pred} by $pass/fail$ label given input scores recorded during the second round of trials (4,000 samples). (a)—predicted output score distributions for the *pass* label. (b)—predicted output score distributions for the *fail* label. (c)—spread of predicted output scores by label.**

The spread of predicted outputs OS_{pred} and OE_{pred} for samples labelled as $pass$ were 0.86 and 0.19, respectively. In contrast, the spread of OS_{pred} and OE_{pred} scores for samples labelled as $fail$ were 0.50 and 0.15, respectively. The distributions of the resultant OS and OE scores post-implementation are shown in Figure 10.9.

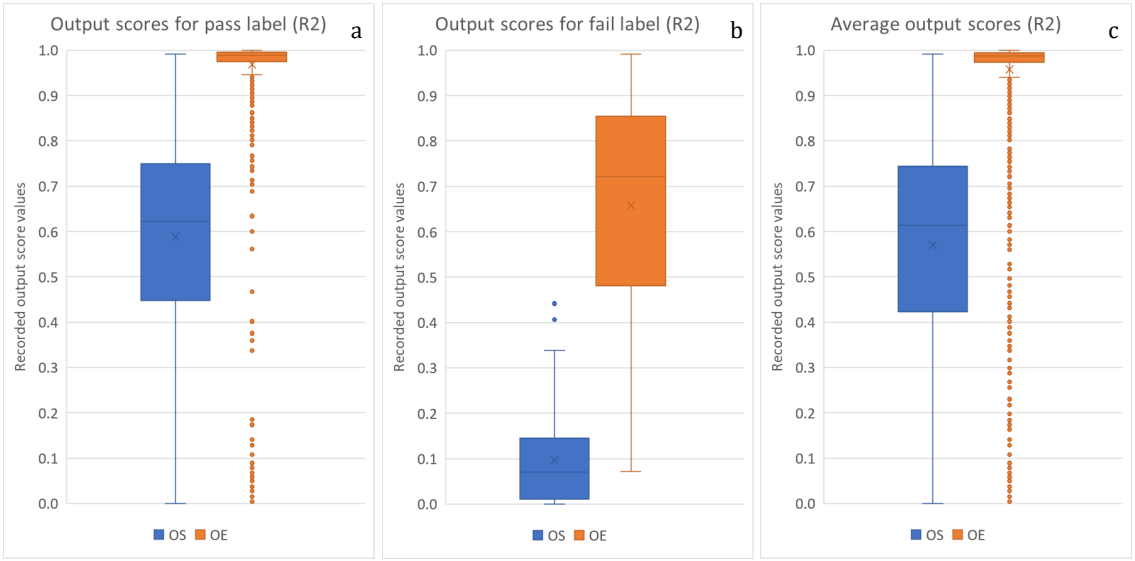


***Figure 10.9*. Distribution of recorded output scores OS and OE by $pass/fail$ label during the second round of trials (4,000 samples). (a)—predicted output score distributions for the $pass$ label. (b)—predicted output score distributions for the $fail$ label. (c)—average output score distributions recorded for the second round of trials.**

As per the STG3_v1 relationship, predicted scores OS_{pred} and OE_{pred} were generally not as polarised as their measured resultants, OS and OE. The predicted distribution more closely resembles the actual outcome. This may be related to the increase in training data as 2,000 samples were used for training in the second round of trials, compared to 500 from trial 1. Moreover, the sensitivity of *OSpredNet_V3* and *OEpredNet_V3* was significantly higher in this trial. The performance in terms of OS and OE for the second round of trials is illustrated in Figure 10.10.

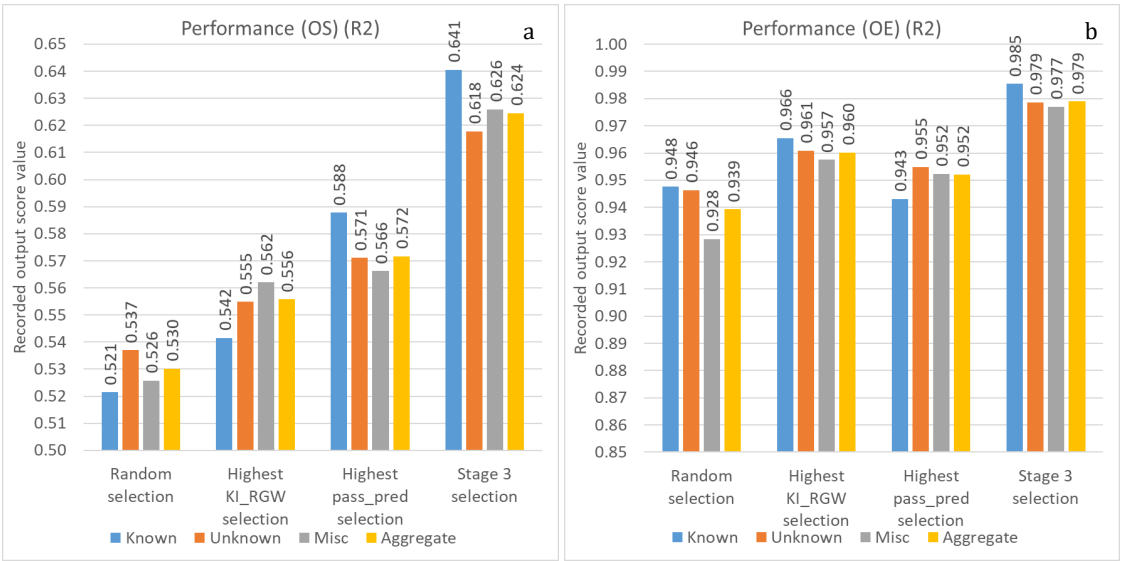


***Figure 10.10*. Measured results from 4,000 grasp attempts by selection criterion and object series. (a)—recorded OS performance. (b)—recorded OE performance.**

Randomly selecting a grasp for implementation from candidate matrix g yielded an average OS score of 0.53 and OE score of 0.94. OS and OE scores were improved slightly to 0.56 and 0.96, respectively, for selection via highest $K_{I_{RGW}}$. Highest $pass_{pred}$ selection yielded an average OS score of 0.57 and OE score of 0.95. Optimising the selection strategy for OS_{pred} and OE_{pred} resulted in OS and OE scores of 0.62 and 0.98, respectively. This improvement in similarity scores was expected and generally resulted in better grasps, qualitatively. In addition to recording the resultant OS and OE scores from 4,000 trials, the related input score matrix S_c values were also recorded, as graphed in Figure 10.11.

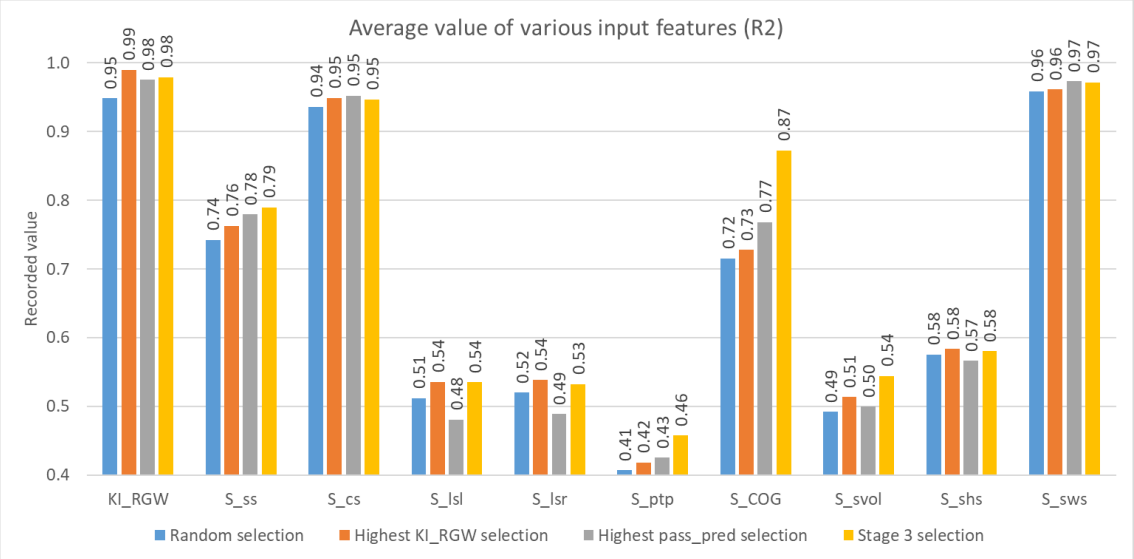


Figure 10.11. **Performance of various input scores S_c by selection criterion, measured from the 4,000 grasp attempts from the second round of trials.**

Generally, grasps selected via highest combined OS_{pred}, OE_{pred} were closer to the COG of an object, compared to other selection criteria. This is reflected by the COG distance score S_{COG}. This behaviour is consistent with the qualitative assessment discussed in Section 10.2. Moreover, grasps selected using the stage 3 component tended to be slightly more symmetric S_{ss}, occupy more area in the grasping window S_{ptp} and occupy more area from the side-view perspective S_{svol}. For a comprehensive description of the scores within matrix S_c, the reader is directed to Section 4.2. Interestingly, stage 3 selection did not prioritise the grasp classification confidence score, quantified by the stage 2 softmax output $K_{I_{RGW}}$. This behaviour is ideal as a high-quality grasp is not necessarily the top-scoring grasp from the perspective of the gripper. OS and OE scores recorded in the second round of trials by series, category and label are illustrated in the form of histograms in Figures 10.12 and 10.13.

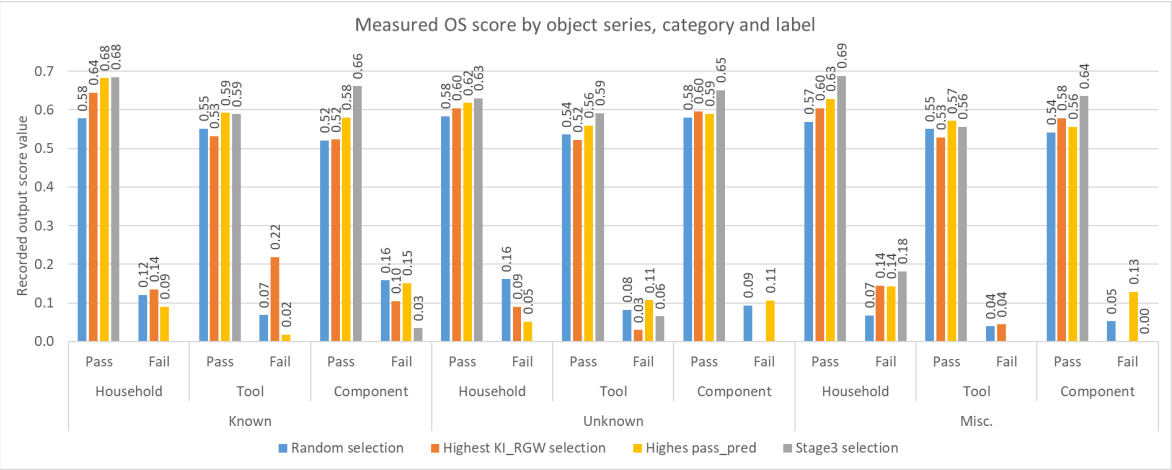


***Figure 10.12*. Average *OS* scores recorded by object series, category, label and selection criterion. Note that some values are not present as no grasp failures were recorded in some circumstances.**

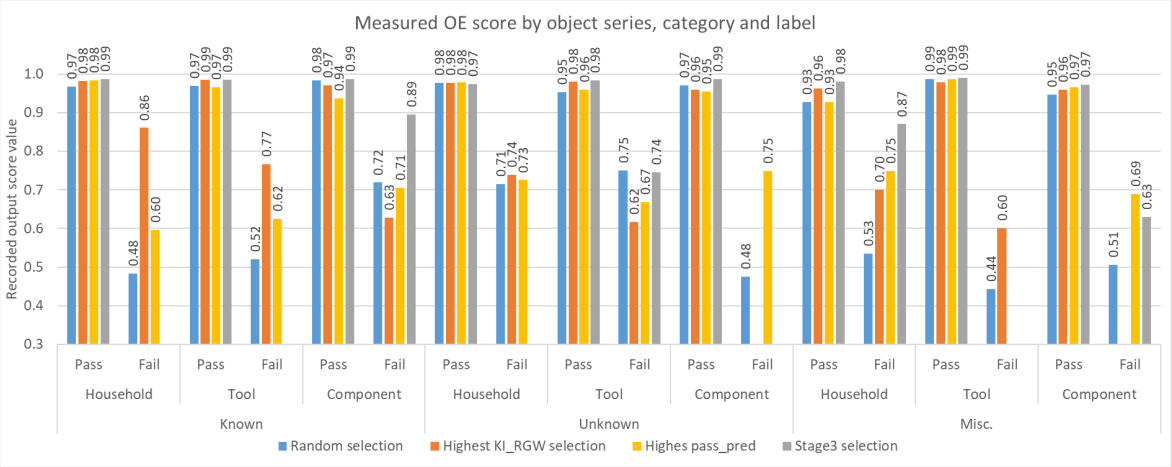


***Figure 10.13*. Average *OE* scores recorded by object series, category, label and selection criterion. Note that some values are not present as no grasp failures were recorded in some circumstances.**

OS and *OE* performance by $pass/fail$ label and selection criterion are illustrated in Figure 10.14.

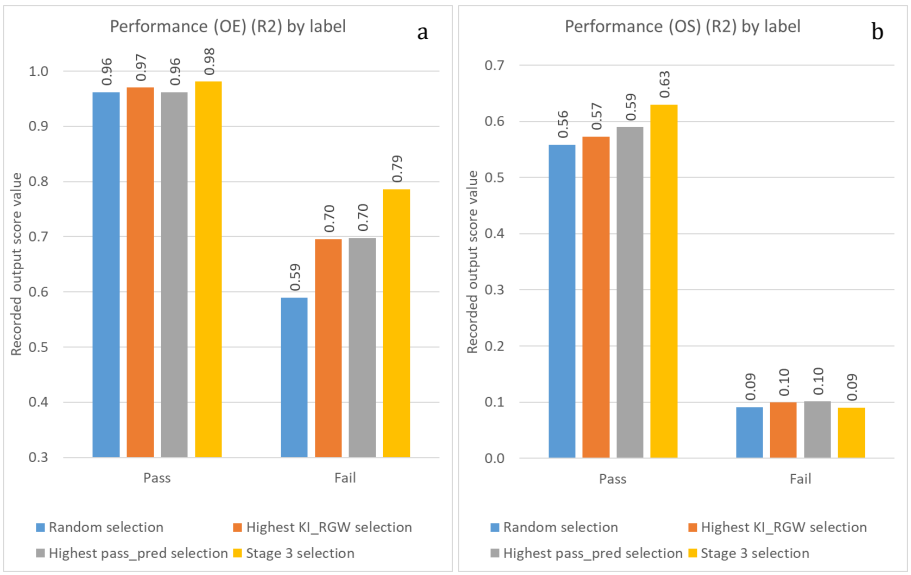


***Figure 10.14*. Measured output score value by $pass/fail$ label from 4,000 grasp samples. (a)—average OS score by selection criterion and label. (b)— average OE score by selection criterion and label.**

From the 4,000 grasp attempts recorded in the second round of trials, it was clear that the quality of a grasp—quantified by OS and OE—improved by selection criterion (Figure 10.10). Compared to the random selection baseline, grasping via stage 3 selection improved OS by 9.4%. For the orientation score OE, this equates to an improvement of 4.0%. As expected, the highest $K_{I_{RGW}}$ and $pass_{pred}$ selection criteria yielded higher OS and OE scores compared to random selection, but lower scores than highest combined OS_{pred}, OE_{pred}. This pattern is also shared by the aggregate grasp rate illustrated in Figure 10.6. Depending on the selection criterion, unsuccessful grasps tended to fail such that object orientation was less affected, quantified by OE. For grasps within the $fail$ label, this equates to an OE disparity of 19.6% when stage 3 selection is employed, relative to random selection. Therefore, as the quality with which grasps are handled increases, the quality with which grasps fail also increases. However, this relationship was not present within OS for grasps labelled as $fail$. The average OS and OE measurements by $pass/fail$ label from round 2 are illustrated in Figure 10.15.

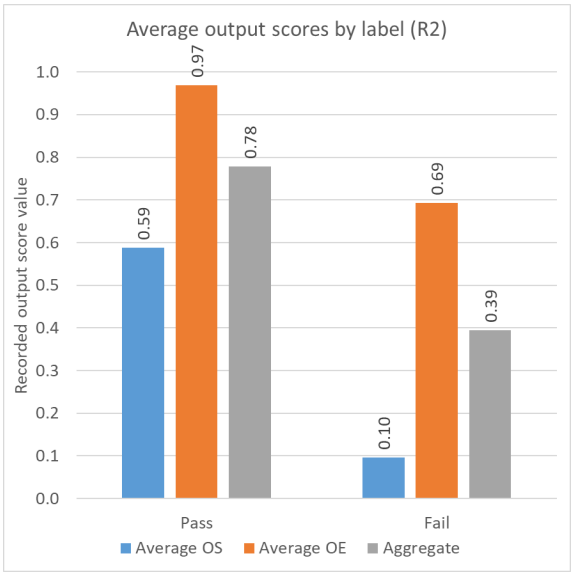


***Figure 10.15*. Average output score by $pass/fail$ label from the second round of trials (4,000 samples).**

The average OS and OE scores for a grasp trial labelled as $pass$ were 0.59 and 0.97, respectively. In contrast, the average OS and OE scores for a grasp trial labelled as $fail$ were 0.10 and 0.69, respectively. This outcome is very similar to the outcome recorded in the STG3_v1 dataset (Figure 10.4) and supports the implication that there may be a relationship between OS, OE and the literature-based outcome, $pass/fail$.

In addition to grasp rate and quality, the time taken to generate and select a grasp are key aspects related to the proposed methodology. The computation time for each major component at trial time is shown in Table 10.4.

***Table 10.4*. Measured run-time per component.**

Component	Time to compute
Object segmentation	$4.7 \times 10^{-2}\ s$
Stage 1 network	$3.7 \times 10^{-3}\ s$
Sobel operation	$7.9 \times 10^{-2}\ s$
Stage 2 network	$3.8 \times 10^{-4}\ s$
Stage 3 network (OS_{pred})	$7.9 \times 10^{-5}\ s$
Stage 3 network (OE_{pred})	$4.9 \times 10^{-5}\ s$
Overall system	$5.28\ s$

The overall system time cited in Table 10.4 relates to the time taken to image the object, generate candidate grasps and select a final grasp for implementation, and includes hardware acquisition times, e.g., camera frame capture, load-cell measurements, communication, etc. Note that these are the computation times for high-resolution settings used during trials. On average, the proposed methodology produced 82.7 candidate grasps of $K_{I_{RGW}} \geq T_{grasp}$ per object with i_{step}, j_{step} and $y_{s_{step}}$ resolutions of 10, 20 and 20, respectively, when considering 4 Sobel rotations θ_{Sobel}. These step magnitudes resulted in an operating frequency of approximately 0.2 Hz. i_{step}, j_{step} and $y_{s_{step}}$ and the number of Sobel operations θ_{Sobel} were the largest methodology-based contributors to computation time. Increasing step magnitudes—thus lowering resolution—significantly increased computation time but reduced the total number of candidate grasps generated.

10.2 Qualitative analysis

During physical trials, many implemented grasps qualitatively considered by the operator as *poor* were labelled as $pass$ by the manipulation procedure noted in Section 9.4, although the resultant OS and OE scores were generally lower for such grasps. Grasps deemed poor usually suffered from several sources of grasp-induced error, e.g., an inappropriate gripper-object interface, significant object displacement caused by the gripping process and droop. A poor grasp did not necessarily result in a dropped object and was therefore not reflected by the literature-based $pass/fail$ metric. Poor grasps associated with the $pass$ label generally yielded OS and OE scores lower than 0.30 and 0.50, respectively. Grasps considered as *excellent* tended to result in OS and OE values higher than 0.80 and 0.98, respectively.

Qualitatively, it was observed that the employed selection criterion significantly impacted which grasp was implemented from candidate matrix g. Highest $K_{I_{RGW}}$ tended to select grasps better suited to the gripper, while highest combined OS_{pred}, OE_{pred} gravitated toward grasps close to the COG of an object—though well-suited grasps were also selected.

As expected, no obvious pattern was observed for random selection. For graphical examples illustrating the various behaviours by selection criterion, see Figures 10.16-26.

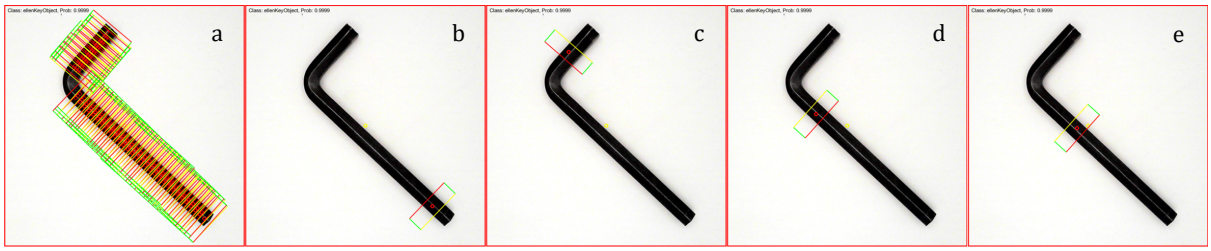


***Figure 10.16*. Large hex key (kT003). (a)—183 candidate grasps g generated for this object. (b)—grasp selected via random selection. (c)—grasp selected via highest $K_{I_{RGW}}$ selection. (d)—grasp selected via highest $pass_{pred}$. (e)—grasp selected via highest combined OS_{pred}, OE_{pred}.**

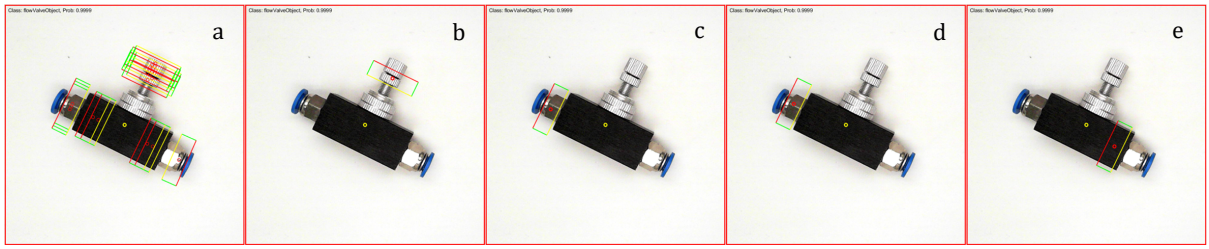


***Figure 10.17*. XCPC pneumatic valve (kC005). (a)—21 candidate grasps g generated for this object. (b)—grasp selected via random selection. (c)—grasp selected via highest $K_{I_{RGW}}$ selection. (d)—grasp selected via highest $pass_{pred}$. (e)—grasp selected via highest combined OS_{pred}, OE_{pred}.**

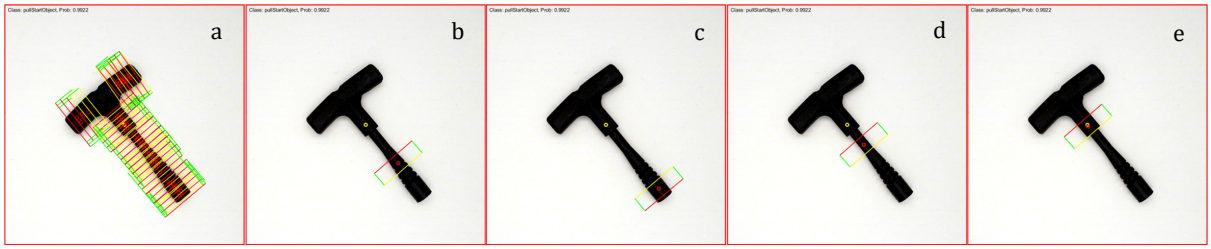


***Figure 10.18*. Pull start handle (kC003). (a)—71 candidate grasps g generated for this object. (b)—grasp selected via random selection. (c)—grasp selected via highest $K_{I_{RGW}}$ selection. (d)—grasp selected via highest $pass_{pred}$. (e)—grasp selected via highest combined OS_{pred}, OE_{pred}.**

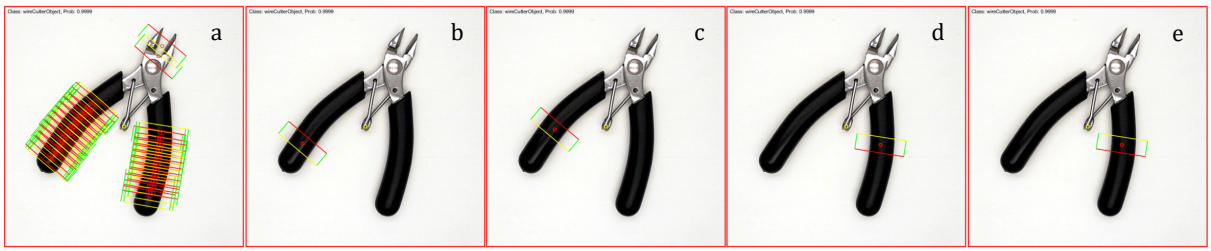


***Figure 10.19*. Small side cutters (kT005). (a)—101 candidate grasps g generated for this object. (b)—grasp selected via random selection. (c)—grasp selected via highest $K_{I_{RGW}}$ selection. (d)—grasp selected via highest $pass_{pred}$. (e)—grasp selected via highest combined OS_{pred}, OE_{pred}.**

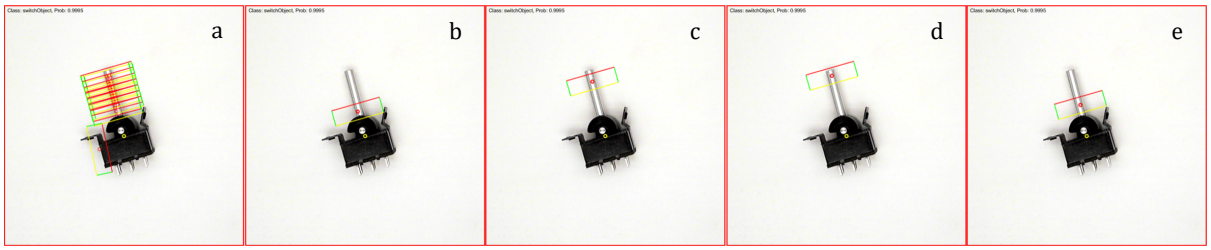


***Figure 10.20*. Electrical switch (kT001). (a)—25 candidate grasps g generated for this object. (b)—grasp selected via random selection. (c)—grasp selected via highest $K_{I_{RGW}}$ selection. (d)—grasp selected via highest $pass_{pred}$. (e)—grasp selected via highest combined OS_{pred}, OE_{pred}.**

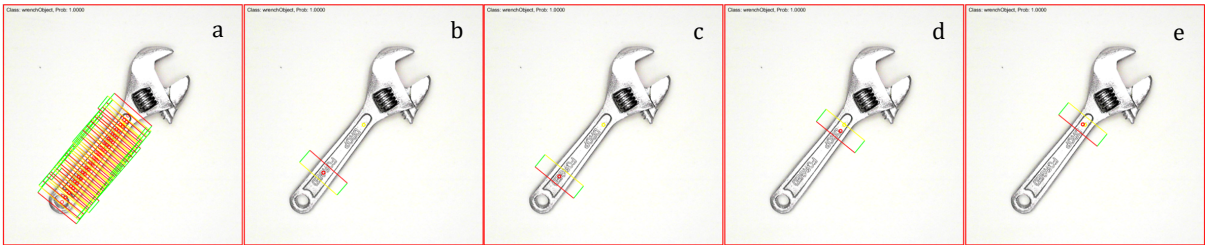


***Figure 10.21*. Small adjustable wrench (kT004). (a)—75 candidate grasps g generated for this object. (b)—grasp selected via random selection. (c)—grasp selected via highest $K_{I_{RGW}}$ selection. (d)—grasp selected via highest $pass_{pred}$. (e)—grasp selected via highest combined OS_{pred}, OE_{pred}.**

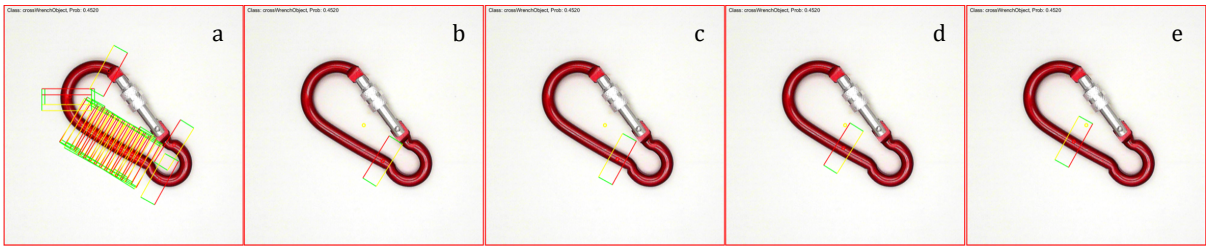


***Figure 10.22*. Red carabiner (mT004). (a)—61 candidate grasps g generated for this object. (b)—grasp selected via random selection. (c)—grasp selected via highest $K_{I_{RGW}}$ selection. (d)—grasp selected via highest $pass_{pred}$. (e)—grasp selected via highest combined OS_{pred}, OE_{pred}.**

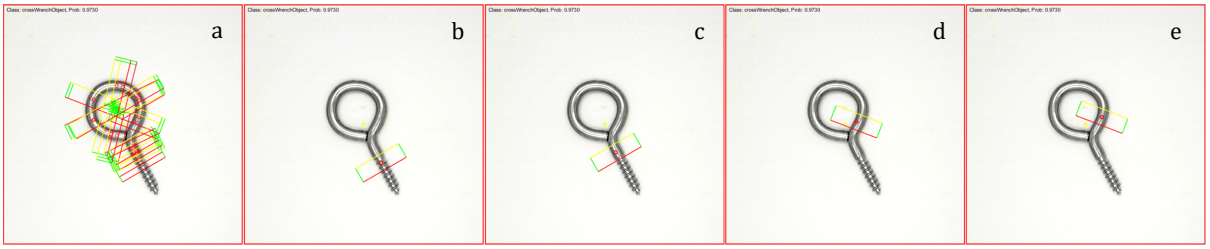


***Figure 10.23*. Silver eyelet screw (uC013). (a)—34 candidate grasps g generated for this object. (b)—grasp selected via random selection. (c)—grasp selected via highest $K_{I_{RGW}}$ selection. (d)—grasp selected via highest $pass_{pred}$. (e)—grasp selected via highest combined OS_{pred}, OE_{pred}.**

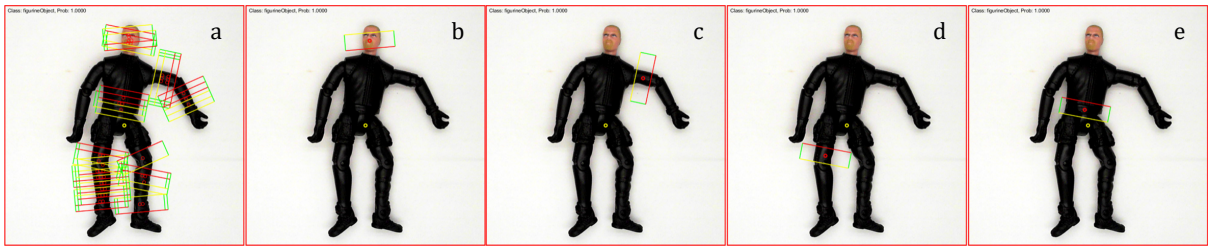


***Figure 10.24*. Figurine (kH002). (a)—44 candidate grasps g generated for this object. (b)—grasp selected via random selection. (c)—grasp selected via highest $K_{I_{RGW}}$ selection. (d)—grasp selected via highest $pass_{pred}$. (e)—grasp selected via highest combined OS_{pred}, OE_{pred}.**

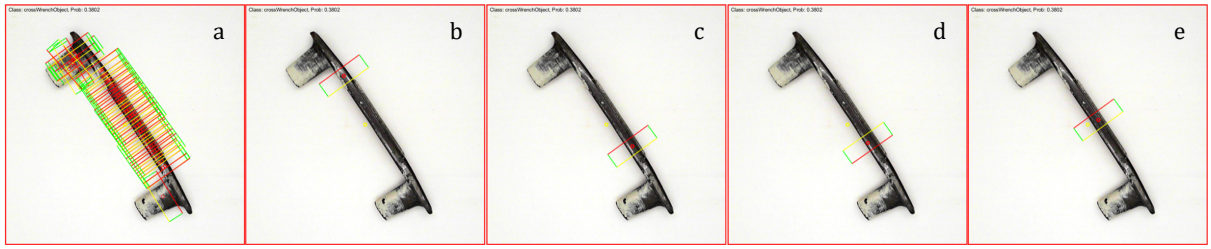


***Figure 10.25*. Door handle (mC013). (a)—96 candidate grasps g generated for this object. (b)—grasp selected via random selection. (c)—grasp selected via highest $K_{I_{RGW}}$ selection. (d)—grasp selected via highest $pass_{pred}$. (e)—grasp selected via highest combined OS_{pred}, OE_{pred}.**

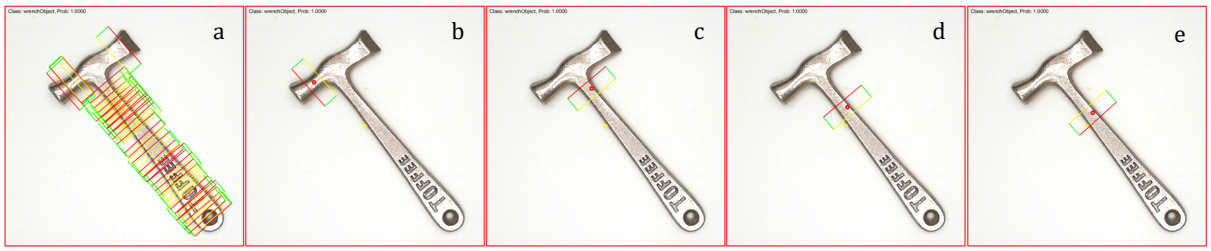


***Figure 10.26*. Walkers nutcracker (uT004). (a)—129 candidate grasps g generated for this object. (b)—grasp selected via random selection. (c)—grasp selected via highest $K_{I_{RGW}}$ selection. (d)—grasp selected via highest $pass_{pred}$. (e)—grasp selected via highest combined OS_{pred}, OE_{pred}.**

Rectangular grasping windows I_{RGW} with a high likelihood of resulting in poor OS and OE scores, or fail the grasp altogether, were largely distinct from the average sample. The left-most, vertically aligned grasp in Figure 10.20—a for instance, would be considered a *poor* candidate on account of an inappropriate gripper interface. The subsequent grasp selected from this pool via random selection (Figure 10.20—b) has the potential for gripper collision, resulting in poor grasp quality, if not a $fail$ outcome. However, it has been observed that an object may simply be displaced in such cases as the gripper is lowered into position, generally resulting in low OS and OE scores, but a successful grasp. The two candidate grasps around the cutting blades of the small side cutter in Figure 10.19—a would also be considered *poor*, as the object-interface is not obviously well-suited, and this part of the object is not rigid.

Many samples selected via random selection, highest $K_{I_{RGW}}$ or highest $pass_{pred}$ performed poorly due to low COG distance scores S_{COG}. As the distance between I_{RGW} centre x_t, y_t and COG position $x_{t_{COG}}, y_{t_{COG}}$ increases, gripper closure force and contact area become increasingly important to mitigate object droop, which may be defined as the vertical displacement of an object during handling once it has already been grasped, due to gravity. This was primarily relevant for heavier objects, e.g., large hex key (kT003), small adjustable

wrench (kT004), small side cutters (kT005), motorcycle chain (uC001), small pliers (uT008), Walkers nutcracker (uT004), medium hex key (mT001). The wrench object (kT004) illustrated in Figure 10.21 was particularly prone to droop due to its uneven mass distribution, i.e. more mass was concentrated toward the adjustable end of the object. As such, this object tended to droop when not grasped close to the COG. Highest $K_{I_{RGW}}$ tended to select grasps toward the handle (Figure 10.21—c), usually resulting in lower OS and OE scores. In contrast, highest combined OS_{pred}, OE_{pred} selection tended toward the COG of the object (Figure 10.21—d)—avoiding droop, thereby improving grasp quality. Generally, grasps toward the handle did not result in failed grasp attempts, although OS and OE were severely affected. Similarly, significant droop was observed for the XCPC pneumatic valve object (kC005) when grasped at the metallic knob, shown in Figure 10.17—b. Moreover, a lack of friction was observed when grasping the metallic fitting section, shown in Figure 10.17—c. However, grasps for this object were very secure when implemented around the body (Figure 10.17—d). Grasp-COG distance was also of significant importance for the large hex key (kT003) (Figure 10.16), Walkers nutcracker (uT004) (Figure 10.26) and the door handle (mC013) (Figure 10.25).

Overall, grasps selected via the stage 3 component were qualitatively preferable. False positives, non-parallel grasps and generally poor candidates produced by the *graspNet_V4_15* network were usually avoided by this mode of selection. Furthermore, grasping objects closer to their COG tended to result in more secure grasps—reflected by OS and OE. In addition to qualitatively observable patterns of selection by criterion, several common reasons for poor quality grasps were observed. Figure 10.27 illustrates some implemented grasps associated with the *pass* classification, but low OS and OE scores.

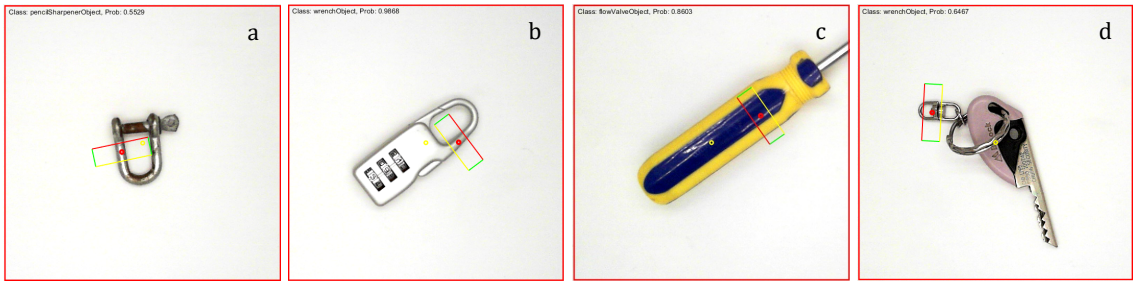


***Figure 10.27*. Various selected grasps resulting in a *pass* label, but poor OS and OE scores. (a)—small D-clamp object (uC012). (b)—silver combination lock (uT015). (c)—blue and yellow screwdriver (uT005). (d)—key object (uH006).**

Rectangular grasping windows I_{RGW} do not capture the physical dimensions of the gripper outside of this window. Therefore, the trialled selection strategies—random, highest $K_{I_{RGW}}$, highest $pass_{pred}$ and highest combined OS_{pred}, OE_{pred}—have no context of the surrounding area. In some extreme cases, this led to grasps that resulted in collisions with the gripper. Figure 10.27—a and 10.27—b illustrate I_{RGW} windows that may result in collisions with the gripper due to lack of context. As a result of the shape of the gripper fingers, such grasps generally displaced the object prior to force-closure, thus decreasing OS and OE but maintaining a *pass* classification. In other cases, grasping areas were selected that barely fit within the maximum opening width of the gripper. Though this is not problematic itself, it did exacerbate other issues. Figure 10.27—c illustrates a situation in which the object width is only slightly smaller than the maximal gripper opening. The candidate window I_{RGW} is also off-centre. In such cases, significant displacement was observed as the gripper was lowered onto the object—forcing parallel alignment.

The key object (uH006) was exceptionally difficult to grasp. Many candidate grasps were centred around the key part of the object, resulting in appropriate implementation. However, candidates were not uncommonly situated around the linkage shown in Figure 10.27—d. When grasped by this linkage, the configuration of the object changed drastically—as the object was suspended freely when lifted. When placed back on the conveyor, object orientation and position were considerably affected, resulting in very low OS and OE scores. Note that the key object was not usually dropped when grasped by this location, though the potential for useful manipulation was very poor. Several common modes of failure were observed during the second round of trials. Some examples of selected grasps that resulted in a $fail$ outcome, and generally very low OS and OE scores, are illustrated in Figure 10.28.

***Figure 10.28*. Various selected grasps resulting in a $fail$ label and generally very poor OS and OE scores. (a)—blue dragonfly toy (uH011). (b)—brown scorpion toy (uH007). (c)—blue Officemax pen (mT005). (d)—fixed figurine (mH004). (e)—key (uH006). (f)—PK chewing gum (mH010). (g)—rod end bearing (kC002). (h)—silver combination lock (uT015). (i)—red sealer (mH009). (j)—XCPC pneumatic valve – no fittings (mC008). (k)— XCPC pneumatic valve – no fittings (mC008). (l)—pneumatic T-splitter (kC004).**

Deformable objects seldom caused grasps to fail but affected OS and OE scores. Objects considered significantly deformable include the red Euoplocephalus toy (mH008), brown scorpion toy (mH007), figurine (kH002) and blue Plesiosaur toy (uH014). The blue dragonfly toy (uH011) presented a unique case due to its transparent wings (Figure 10.28—a), in which an incorrectly selected window could result in the grasp action failing to close onto the object altogether—consequently failing to lift the object. In such circumstances, trials were labelled as $fail$. Such grasps displaced the gripper upwards, simply exerting a downward force on the object. Therefore, the object was not disturbed by the grasp attempt—resulting in high OS and OE measurements. Usually, candidate grasps were centred around the tail end of this object.

Due to the vibration caused by the conveyor stepper motor during operation, it was observed that very light and rigid objects were slightly displaced relative to the imaged location in I_t when translating the object between robot and vision spaces. The error introduced by vibration did not account for any failed grasps from the 4,000 trials, but marginally reduced OS and OE measurement accuracy in some extreme cases.

The largest contributor to failed grasps was the selection of samples incorrectly classified by the stage 2 component. Such samples often caused major collisions (Figure 10.28—d, f, g, h, j, k, l) or translated the object during the grasp attempt (Figure 10.28—b, c, e, i). Without force-closure, an object could not be lifted, thus resulting in a $fail$ label and poor OS and OE scores. Poor sample quality was reflected in the score matrix S_c. Therefore, incorrectly classified samples were generally avoided by the stage 3 selection process.

10.3 General discussion

Computation time of the proposed methodology at trial time was 5.3 seconds. The time taken to capture 2 images, approximately 0.067 seconds, and other hardware processes are included in the cited computation time. The time taken to compute a final grasp fluctuates significantly throughout literature, ranging from 19 milliseconds [72], to approximately 3 seconds [62] or even 30 seconds [16]. The methodology proposed by Sainul, Deb and Deb took between 4 and 23 seconds to compute a grasp. Their method was also implemented in Matlab but utilised a slicing-based planner. The work by Lenz, Lee and Saxena [13] and Sun, Yu, Liu and Gu [66] most closely resembles the grasp synthesis methodology proposed in this thesis. Their works were implemented in Matlab, utilised similar computer hardware and were based on the generate-and-test structure. Both works generated their final grasp for implementation after approximately 14 seconds. The computer hardware utilised in this work is cited in Table 9.1, Chapter 9.2. Note that this hardware is relatively low-end compared to contemporary specifications.

The cited time of 5.3 seconds was measured with low step values employed, thus, greatly expanding the overall grasp search space at the cost of computation time. Increasing $y_{s_{step}}$ to 50 resulted in an overall image-grasp time of 4.2 seconds, while still maintaining an average candidate matrix g size of 83. Note that this resulted in reduced robot z-axis resolution $\bar{z}$. Reducing the number of Sobel operations θ_{Sobel} to 2, cut this time to 2.66 seconds and resulted in approximately 55 candidate grasps. Limiting the number of Sobel operations θ_{Sobel} considered to 1 reduced this time to 1.6 seconds, generally resulting in 44 grasp candidates.

Though higher resolution settings were used during trials, it was found that more pragmatic step magnitudes could be used during general operation without significantly impacting the resultant OS and OE scores—referred to as the general operating mode. An operating resolution of $i_{step} = 15$, $j_{step} = 40$, $y_{s_{step}} = 50$ and 4 Sobel operations generated an average of 55 candidate grasps within 1.75 seconds. 4 Sobel operations enabled the search for candidate grasps at 4 distinct angles. Generally, the selected grasp G belonged to the Sobel angle related to the major orientation of the object, i.e. the highest number of pixels counted at a specific angle. Therefore, reducing this value did not usually affect the grasp selected for implementation for simpler objects. However, this value greatly impacted the search resolution for more complex objects. Reducing the number of Sobel operations to 2 reduced computation time to 0.7 seconds and the total g to 19. Assessing grasps at only 1

Sobel rotation resulted in a computation time of 0.5 seconds and generally produced 10 candidate grasps.

An increase in step values $i_{step}, j_{step}, y_{s_{step}}$ and a reduction in the number of Sobel angles considered θ_{Sobel} improves computational performance by reducing the number of classification windows I_{RCW} iterated and assessed throughout rotated space $I_{BBR_{1 \,..\, 4}}$. Consequently, the final number of candidate grasps within g is decreased. For more information related to grasp synthesis, refer to Chapter 4.3. This lower resolution search space results in the sampling of less potential grasp areas. Thus, the *ideal* candidate might not be considered by the stage 3 selection component. The resolution inherent to the top-view image space I_t is relatively high compared to the conveyor belt system error ($\pm$ 3.4 pixels), robot-vision transformation accuracy ($\pm$ 1.70 pixels) and manipulator repeatability ($\pm$ 1.3 pixels). Therefore, the error introduced by selecting samples that slightly deviate from the *ideal* may not necessarily impact OS and OE in practice, provided step magnitudes are limited appropriately. However, the additional redundancy provided by the number of Sobel operations was found to be crucial when generating grasps for more complex geometries.

Testing with the mid-range hardware listed in Table 10.5 reduced computation time by a factor of 5, resulting in a high-resolution operating frequency of approximately 1 Hz and an operating frequency of approximately 3 Hz in general operating mode. Generally, the wider community considers Matlab an excellent platform for research and development. For real-world applications however, better performance is usually associated with Python-based implementation.

***Table 10.5*. Mid-range computer hardware details used for testing.**

Component	Details
Graphics card	NVIDIA GeForce RTX 2070 [416] • 8GB GDDR6 • Compute capability: 7.5 • Memory bandwidth: 448 GB/s • Power usage: 185 W
CPU	AMD Ryzen 7 2700 [417] • 8 cores, 16 threads • 3.20 GHz (4.10 GHz turbo boost) • TDP: 65 W
RAM	16 GB • 2,660 MHz

During the initial stages of development, it was observed that the behaviour of the stage 2 component varied somewhat between networks, even when identical data was used during training. Some networks for instance, failed to generate any candidate grasps of $K_{I_{RGW}} > T_{grasp}$ in some cases, or exhibited higher softmax outputs $K_{I_{RGW}}$ for candidates qualitatively considered poor. Generally, such issues were reflected in the associated confusion matrix and did not persist as the number of training samples increased. Mitigation of qualitatively poor rectangles I_{RGW} was particularly noticeable between networks trained using the STG2_v2 and STG3_v3 datasets and may be a result of the additional samples and skew toward the negative class present in STG3_v3. The learning component for grasp selection is detailed in Section 7.2.

The orientation score OE is computed by comparing the pre-grasp angle θ_{pre}, with the post-grasp angle θ_{post}. The derivation of these angles is based on the major axis of an ellipse fitted to the object—noted in Chapter 9.3.6. Circular objects do not have a major orientation. Therefore, OE cannot be computed for such objects. In practice, the sensitivity of the vision system captured and responded to irregularities within seemingly circular objects. Although this issue was not found relevant in this study, it may be problematic in some other cases. Moreover, OE is computed from perspective I_t. As such, changes in object geometry from other perspectives are not necessarily captured by this metric, as illustrated by Figure 10.29.

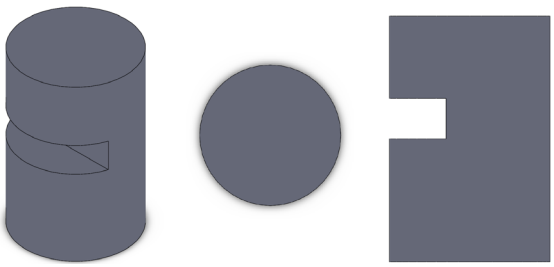


***Figure 10.29*. Illustration of object geometry that may not be captured by OE metric from perspective I_t (middle picture).**

Circular objects also posed a challenge to the Sobel operation. This operation computes the gradient angle of each pixel within I_{BB} and utilises the resulting statistical histogram to find significant angles within the given image. Circular objects do not have significant angles or edges. Therefore, the resulting histogram does not produce well-defined peaks. Figure 10.30 illustrates the varying response from differing geometries.

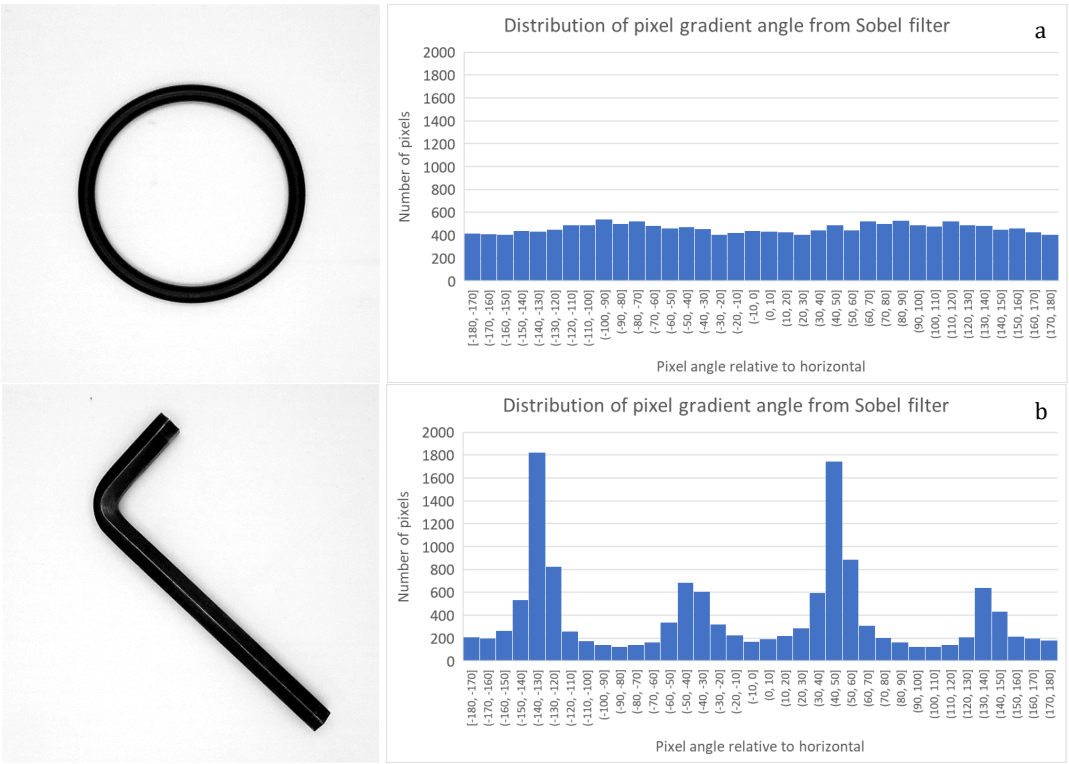


***Figure 10.30*. Illustration of the statistical histograms resulting from unique object geometries. (a)—circular object and the resultant statistical histogram. Note that the histogram distribution is approximately flat. (b)—sharp-edged object and the resultant statistical histogram. Note the strong peaks present in this distribution.**

A lack of well-defined peaks within a resultant Sobel histogram, i.e. no major object orientation, did not contribute to failed grasp attempts or reduce resultant OS and OE scores. From the trials conducted in this thesis, it was concluded that the orientation of the selected grasp does not negatively influence the resulting trial for objects with no measurable major orientation. Example grasps for such objects are shown in Figure 10.31.

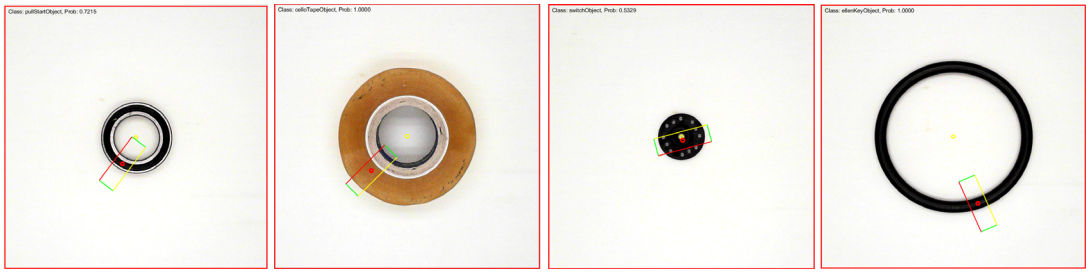


***Figure 10.31*. Illustration highlighting the inconsequential effect of grasp angle for approximately circular objects.**

Four separate selection criteria were tested during trials: random selection, highest $K_{I_{RGW}}$ selection, highest $pass_{pred}$ and highest combined OS_{pred}, OE_{pred} selection. Randomly selecting a grasp for implementation from candidate grasp matrix g was straightforward, utilising the built-in random number generator provided by Matlab. Similarly, selections made using $pass_{pred}$, OS_{pred} and OE_{pred} were based on the continuous outputs provided by regression networks. Highest $K_{I_{RGW}}$ selection proved problematic in some cases, as several samples from g may be classified with a stage 2 softmax output value of 1.0. Therefore, the highest $K_{I_{RGW}}$ selection strategy could not distinguish between such samples and defaulted to random selection within this subset. This issue may be partly responsible for the performance related to this selection criterion and supports the utility of more sophisticated selection strategies—such as the grasp selection component proposed by this thesis.

As explored in Chapter 2.6, it is difficult to determine the relative performance of this research due to the lack of clear benchmarks, experimental protocols and shared metrics. However, many works subscribe to the notion of grasp success. In this thesis, a grasp is labelled as either a $pass$ or $fail$ based on the manipulation procedure outlined in Section 9.4. This process results in a $pass$ label if the object is lifted vertically to a height of 15 cm, held stationary for 10 seconds, then placed back on the conveyor. If the object falls at any point during this action, the trial is labelled as $fail$. The derivation of this definition stems from previous works throughout literature. Saxena *et al.* for instance [1, 59], consider a trial successful if an object can be lifted to a height of 1 foot and held stationary for 30 seconds. Similarly, Pinto and Gupta [60], Johns, Leutenegger and Davison [42] and Kopicki *et al.* [81] use the same standard, but substitute a height of 20 cm. Many other works also consider a grasp attempt successful if the trialled object is lifted above some pre-defined height for some time interval [36, 51, 52, 72].

To help normalise the comparison with other works, common *household* and *laboratory* items that are similar in shape and size to objects used by methodologies reported in literature and within the capability of the Dobot Magician were selected. Moreover, consideration centres on 2-fingered, vision-based works that physically trialled their approach and made available information related to trials, their object test pool and datasets. Table 10.6 lists the range of objects and number of experimental trials reported by various other approaches.

Table 10.6. **Cited details related to the employed object test pool and number of trials conducted for various works throughout literature. Methodological approach and testing conditions varied.**

Author	Number of objects in test pool	Description	Total number of trials
Saxena *et al.* [59]	16	Common household objects (6 similar to training data, 10 novel)	52
Saxena *et al.* [86]	13	Common household objects (novel)	150
Le *et al.* [87]	8	Common household objects (novel)	160
Klingbeil *et al.* [24]	6	Common household objects (novel)	120
Jiang *et al.* [65]	12	Common household objects (novel)	120
Kopicki *et al.* [81]	18	Common household objects (4 used to generate data, 10 used for testing only, 4 float objects)	52
Sun *et al.* [66]	15	Common household objects (known + novel)	150
ten Pas *et al.* [62]	48	Common household objects (18 used to generate samples, 30 used for testing only.	90
Pinto *et al.* [60]	-	Common household objects (known + novel)	150
Johns *et al.* [42]	20	Common household objects (novel)	100
Mahler *et al.* [40]	48	Common household objects (8 used to generate data, 40 used for testing only)	1,000 (various tests)
Viereck *et al.* [75]	10	Common household objects (novel)	40
Chu *et al.* [70]	10	Common household objects (novel)	200
Kalashnikov *et al.* [36]	69	Common household and tool objects (novel—some repeated objects, different colour)	714
Morrison *et al.* [72]	20	Common household objects (novel)	200
Zelenak *et al.* [53]	39	Common household objects (novel)	39
Gualtieri *et al.* [84]	25	Common household objects (novel)	579
Paolini *et al.* [322]	-	Cylindrical objects only (novel)	1,500

The number, range, difficulty and redundancy of objects considered during testing varies significantly. Morrison, Corke and Leitner [72] for instance, utilise a set of common items in addition to 3D printed objects with adversarial geometry—intended to confound the grasping task. Le, Kamm, Kara and Ng [87] only consider 8 objects, 3 of which belong to the same category of object. Commonly, a set of *household* or *everyday* objects is utilised as the test set. Items considered as *household* or *everyday* may include mugs, flashlights, bulbs, bottle caps, cereal boxes, pens, markers, remotes, cans, bottles, controllers, headphones, balls, bowls, tape measures, toys, plush toys, toothbrushes, screwdrivers, side cutters, scissors, staplers, glasses, shoes, groceries, figurines, drink bottles, hammers, sponges, marbles, discs, mobile phones, office supplies, foam ear plugs, books, wooden blocks, etc.

Usually around 100-150 trials are conducted with such objects, with an even number of trials dedicated to each unique object. Though many works report a relative low number of trials or limited set of test objects, many have ulterior goals or try to demonstrate some other criteria they consider indicative of the desired performance. Kalashnikov *et al.* [36] for instance, illustrate several grasping strategies discovered by their deep reinforcement policy. Although their trials consist of 102 grasps per 7 robotic manipulators, their work draws from the experience of over 580,000 physical grasp attempts. Similarly, Kopicki *et al.* [81] perform only 52 trials for physical validation, but rigorously analyse their approach in simulation. Morrison *et al.* [72] consider closed-loop grasping in dynamic environments. The tables included in this subchapter represent a low-resolution view of many works by explicitly focusing on some aspects, hence, the reader is urged to further explore the cited literature. Table 10.7 relates to the approach documented in this dissertation.

***Table 10.7*. Details related to the employed object test pool and number of trials conducted in this work, by trial round.**

Trial	Number of objects in test pool	Description	Total number of trials
Trial round 1	28	Common tool, component and household objects (10 known, 18 novel)	940
Trial round 2	100	Common tool, component and household objects (15 known, 85 novel)	4,000

The complete object test pool utilised in this research is covered in detail in Chapter 5. To help diversify the test set, items from three parent categories were selected: general household items, tool items and part-related items. Approximately one third of this set relates to *household* items, while the remaining objects are centred on common tools or componentry associated with industrial automation. Care was taken to curate a test set with minimal redundancy, i.e. differing shape, size, mass, mass distribution and difficulty. The range of objects employed in this thesis is significantly greater than many others reported throughout literature. In addition, diverse object test pools and larger trials yield higher degrees of confidence associated with the resultant estimates. It is important to note that physical trials only provide an estimate of true performance, although, more diverse object pools and more trials provide more robust approximations. Moreover, the proposed methodology uses only 15 objects during training, but is tested using 85 novel objects. This aims to demonstrate the generalised performance of the 3-stage method when grasping objects with no *a priori* knowledge. In addition to the parameters related to physical trials, many works cite their dataset and network accuracy (Table 10.8).

***Table 10.8*. Cited network training details for various works throughout literature. Methodological approach and testing conditions varied.**

Author	Training samples	Training accuracy (%)	Notes
Saxena *et al.* [59], [86]	2,500	94.2	Samples related to 6 objects. Synthetic. Probabilistic model.
Le *et al.* [87]	420	91.4	Manually labelled as *good* or *bad*. SVM classifier.
Jiang *et al.* [65]	~1,500	98.2	Manually labelled samples. 194 pictures labelled with 1-3 *positive* grasps and 5 *negative* grasps.
Sun et al [66]	5,613	96.9	SVM. Cornell Grasp Detection Dataset. Manually labelled *good* or *bad*.
ten Pas *et al.* [62]	6,500	97.8	Derived from 18 training objects. SVM classifier. 3405 *positive* samples, 3095 *negative* samples.
Pinto *et al.* [60]	50,567	79.5	Derived from real grasp attempts from 150 different objects. Unsupervised, reinforcement learning. Samples automatically labelled as *good* or *bad*.
Johns *et al.* [42]	1,000,000	83.8 (varies by method)	1,000 objects, 1,000 training samples per object. Synthetic data. Mesh-based CNN model.
Mahler *et al.* [40]	6,700,000	93.0 (varies by method)	Synthetic data collated from many datasets. CNN model.

Viereck *et al.* [75]	500,000	-	Synthetic data. Generated from object categories similar to objects in test pool. CNN model.
Chu *et al.* [70]	8,019	96.1	5,110 *positive* samples, 2,909 *negative* samples of 885 RGBD images, 244 objects.
Kalashnikov *et al.* [36]	~580,000	-	Reinforcement learning. Real-world grasp attempt samples.

Collecting real-world grasp attempt data is extremely time-consuming and arduous, as evident in the collection process experienced in this research. As a result, many works report the use of proprietary datasets of less than 500 samples, while other larger institutes implement large-scale data collection banks utilising many manipulators over the course of several months. Kalashnikov *et al.* [36] for instance, recorded over 580,000 grasp attempts utilising 7, self-supervised robotic manipulators. Such difficulties can be circumvented through manually labelled datasets, avoiding physical trials altogether. The Cornell Grasp Detection Dataset for instance, consists of a collection of hand-labelled samples associated with a range of objects—similar to the grasp detection dataset proposed in Chapter 7. Alternatively, synthesised data can be generated. Larger datasets are expected for methods that incorporate this approach. Details related to the training data and training accuracy of this work are tabulated in Table 10.9.

***Table 10.9*. Training details documented in this work by trial round.**

Trial	Classification network	Training samples	Training accuracy	Notes
Trial round 1	*97classNet_V3_11*	11,000	97.0%	Object classification, 11 classes
	graspNet_V3_10	33,000	98.0%	11,000 *positive* samples, 22,000 *negative* samples. Manually labelled from 10 objects.
	OSpredNet_V1	500	0.17 RMSE	Physical grasp attempts from 10 objects with measured outcome.
	OEpredNet_V1		0.14 RMSE	
	passpredNet_V1		0.12 RMSE	
Trial round 2	*classNet_V4_16*	80,000	99.5%	Object classification, 16 classes.
	graspNet_V4_15	141,000	99.4%	47,000 *positive* samples, 94,000 *negative* samples. Manually labelled from 15 objects.
	OSpredNet_V3	2,000	0.07 RMSE	Physical grasp attempts from 15 objects with measured outcome.
	OEpredNet_V3		0.14 RMSE	
	passpredNet_V3		0.08 RMSE	

When supervised learning is employed, many works prefer example cases that have been labelled with the expected outcome, thus avoiding physical trials, making the accumulation of larger datasets more practical. Such approaches effectively train a neural network to recognise grasps considered desirable by the human annotator or automatic criteria. Alternatively, some works record the outcome of physical trials. Such trials may be classified into the desired category and used for training. This work makes use of both techniques. The former approach is used for candidate generation and the latter, for selection.

Methodologies from literature often quantify the performance of their approach in terms of the $pass/fail$ metric. Unfortunately, their experimental methods are generally associated with low confidence intervals due to lack of testing. Table 10.10 illustrates the margin of error at 95% confidence associated with various results reported throughout literature.

***Table 10.10*. Confidence associated with the experimental results of various works. Margin of error is calculated at 95% confidence interval, assuming a normal distribution.**

Author	Grasp rate (%)		
	Known	Unknown	Aggregate
Saxena *et al.* [59]	90.0 ± 13.1%	87.5 ± 11.5%	88.5 ± 8.4%
Saxena *et al.* [86]	-	76.0 ± 6.8%	-
Le *et al.* [87]	-	81.9 ± 6.0%	-
Klingbeil *et al.* [24]	-	91.6 ± 5.0%	-
Jiang *et al.* [65]	-	87.9 ± 5.8%	-
Kopicki *et al.* [81]	88.0 ± 15.9%	86.0 ± 11.3%	86.5 ± 9.3%
Sun *et al.* [66]	-	-	86.0 ± 5.6%
Ten Pas *et al.* [62]	-	88.0 ± 6.7%	-
Pinto *et al.* [60]	73.0	66.0	69.5 ± 7.4%
Johns *et al.* [42]	-	80.3 ± 7.8%	-
Mahler *et al.* [40]	93.0 ± 6.0%	80.0 ± 9.0%	-
Viereck *et al.* [75]	-	-	97.5 ± 4.8%
Chu *et al.* [70]	-	-	89.0 ± 4.3%
Kalashnikov *et al.* [36]	-	-	96.0 ± 1.4%
Morrison *et al.* [72]	-	92.0 ± 5.0%	-
Zelenak *et al.* [53]	-	-	82.1 ± 12.0%
Gualtieri *et al.* [84]	-	-	80.8 ± 3.2%

Many works evaluate their methodology with objects very similar to the objects used to generate their training data, e.g., various sized mugs, similar objects of differing colour, differing brands of the same item, etc. Often, the resulting grasp rate is conflated with the grasp rate of novel objects. In this research, such repeats may not be considered novel due to a lack of sufficient variation. It is for this reason that the unknown object series is distinct from the misc. object series—which includes repeats of this nature. For a comprehensive quantitative analysis of the 3-stage methodology, refer to Section 10.1. The trialled grasp rates for known and unknown objects associated with this work are illustrated in Table 10.11.

***Table 10.11*. Confidence associated with the experimental results of this work, by trial round. Margin of error is calculated at 95% confidence interval, assuming a normal distribution.**

Trial	Selection criterion	Grasp rate (%)		
		Known	Unknown	Aggregate
Trial round 1	Random	-	-	95.0 ± 4.3%
	Highest $K_{I_{RGW}}$	94.0 ± 4.7%	98.3 ± 1.9%	96.8 ± 2.1%
	Highest $pass_{pred}$	92.0 ± 5.3%	95.7 ± 3.0%	93.7 ± 2.9%
	Highest combined OS_{pred}, OE_{pred}	99.0 ± 2.0 %	99.4 ± 1.1%	99.3 ± 1.0%
Trial round 2	Random	93.3 ± 4.0%	93.6 ± 2.3%	94.0 ± 1.5%
	Highest $K_{I_{RGW}}$	94.0 ± 3.8%	96.2 ± 1.8%	96.4 ± 1.2%
	Highest $pass_{pred}$	94.7 ± 3.6%	96.4 ± 1.7%	96.3 ± 1.2%
	Highest combined OS_{pred}, OE_{pred}	99.3 ± 1.3%	98.9 ± 1.0%	99.0 ± 0.6%

Note that misc. items are included in the aggregate margin of error calculation for the second round of trials. The confidence intervals associated with the trials conducted in this work demonstrate that the resultant grasp rate estimates are relatively robust. Further, the comparison of this methodology to other open-loop methodologies that grasp a single

object in isolation, shows improvement. An object is considered isolated if it does not touch any other object in the field of view.

The performance of the methodology in this research was framed in terms of the $pass/fail$ metric, in part, for comparison purposes. Many recognise the limitation of this point of comparison and therefore evaluate their approach using the Cornell Grasp Detection Dataset [13]. Some works that utilise this data are listed in Table 10.12.

***Table 10.12*. Cited network accuracy of various works employing the Cornell Grasp Detection Dataset.**

Author	Dataset accuracy
Cheng *et al.* [68]	90.4%
Mahler *et al.* [40]	93.0%
Redmon *et al.* [69]	88.0%
Chu *et al.* [70]	96.0%
Wang *et al.* [71]	82.0%
Kumra *et al.* [58]	89.2%
Sun *et al.* [66]	96.3%
Caldera *et al.* [73]	93.9%

This dataset provides a practical standard which allows for new works to be placed in context with respect to other approaches that also use this dataset. Such a comparison was considered by this research. Unfortunately, it would not be appropriate to test the 3-stage models on the Cornell dataset for several reasons. Since new datasets were defined in this work, and networks were trained accordingly, there are major differences in sample scale, viewing angle, background and rectangular dimensions. The Cornell dataset for instance, contains images that have been captured from varying viewing angles, whereas the datasets in this work specifically pertain to top-down samples. Moreover, the Cornell method was originally aimed at general object handling in more unstructured, domestic settings for goals such as desk clearing and bin packing, while this work aims to precisely manipulate smaller objects within controlled environments. These differences have been embodied in each respective dataset. Since grasps do not need to be as precise for general household tasks, networks trained using the Cornell dataset would not generate candidate grasps appropriate for the proposed selection methodology.

The grasp detection components documented in this thesis would likely perform very poorly on the Cornell set, but a comparison of this nature means very little in terms of the expected physical grasp rate. As discussed in Section 2.6, many works report significant disparities between dataset performance and physical trial performance. In this work for instance, a disparity of 6.1% was observed between the stage 2 rectangle recognition rate and the corresponding random selection trial baseline. Therefore, attaining a network that performs well on the Cornell dataset does not necessarily produce good grasp rates in practice. Many approaches employ alternative methods in addition or parallel to the network trained using this set, often rendering this mode of comparison irrelevant in the context of the resultant manipulation performance.

The studied grasp synthesis method promotes scalability, ease of integration and is not conceptually contingent on the intrinsic hardware parameters or processing procedures documented in this study. Essentially, the stage 2 component directly relates to the implemented gripper. Therefore, this component can be substituted according to the specifications of new hardware. Similarly, the aim of the stage 3 component is to integrate

and utilise various sources of information—as they become available. This topic is further discussed in Section 11.2.

The proposed methodology makes use of the COG location of an object in top-view image space I_t. The COG score S_{COG} is calculated for each sample as a function of the distance between candidate rectangle I_{RGW} centre $[x_t, y_t]$ and COG position $[x_{t_{COG}}, y_{t_{COG}}]$. 4 load-cells are utilised to measure and compute the COG of an object. Subsequently, this configuration constrains the methodology, in that only a single object may be assessed at any one time. If more than one object is present on the conveyor system during this measurement, the average COG position of all objects will be registered instead. Therefore, the applicability of the prototype is limited if a true COG reading is required. However, in most cases the COG of individual objects can be accurately approximated using vision, which is computationally efficient and may be computed with many objects present. It should be noted that estimating COG through vision significantly reduced the overall computation time. For training purposes, as little error as possible is desired, thus, the true COG of an object was used. In practice, vision can simply be implemented by substituting COG position $[x_{t_{COG}}, y_{t_{COG}}]$ for the object centroid position $[x_o, y_o]$. The object centroid represents a COG estimation under the assumption that the mass of the object is uniformly distributed. To fully mitigate this issue, an engineering solution—as opposed to a scientific advance—may be required. COG approximations become problematic for objects with disproportionately distributed mass. Figure 10.32 illustrates an example.

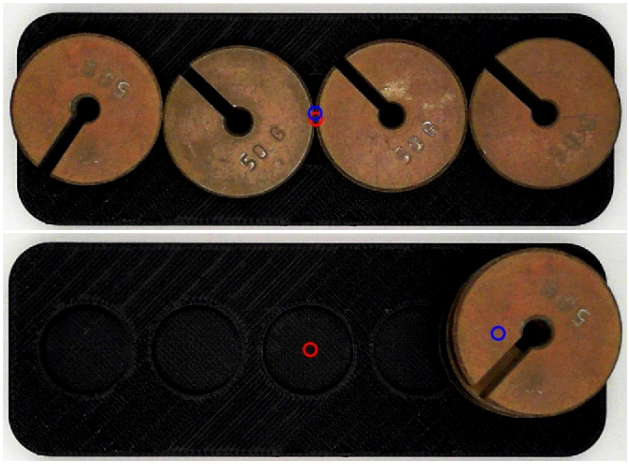


***Figure 10.32*. Illustration of the importance of true COG measurement. The red circle represents the approximated COG of the object through vision. The blue circle represents the COG of an object measured by the load-cell system. (a)—illustration of approximately uniformly distributed object. (b)—illustration of significantly skewed object mass.**

In this work, the COG feature is regarded as one of many potential inputs that provide the selection component with information to improve grasp prediction. It is intended that many varying sensors or known object properties serve as inputs if they are available, e.g., temperature, material properties, friction coefficients, gripper feedback, weight, etc. With the proliferation of IoT sensors, it has become relatively straightforward to measure and incorporate many such properties.

Chapter 11

Conclusions and future work

11.1 Main conclusion

In this research, a 3-stage, sliding-window-based grasp synthesis methodology was proposed that predicts many 2-fingered grasping locations for previously unseen objects using deep learning. Grasp areas considered congruent with the ideal were recorded as candidates. When implementing one such candidate randomly, the physical prototype achieved a grasp rate of 94.0% during trials. Selecting the highest softmax-scoring candidate improved grasp rates to 96.4%. Selecting the candidate with the highest probability of resulting in a $pass$ outcome—as determined by a regression model—achieved a grasp rate of 96.3%. Many works from literature frame the result of a grasp trial as either successful or unsuccessful. Though grasp outcome is a concern, this work argues that grasp quality is equally important, particularly in cases where manipulation and placement quality are essential to the application. Therefore, objective measures of grasp performance are established that comprehensively measure the error introduced by the grasp trial itself, defined in terms of overlap score OS and orientation score OE. The proposed metrics are empirically demonstrated to quantify grasp quality, provide pragmatic network training criteria and offer more descriptive power than traditional measures. By training to optimise for OS and OE, the trialled grasp rate was improved to 99.0%.

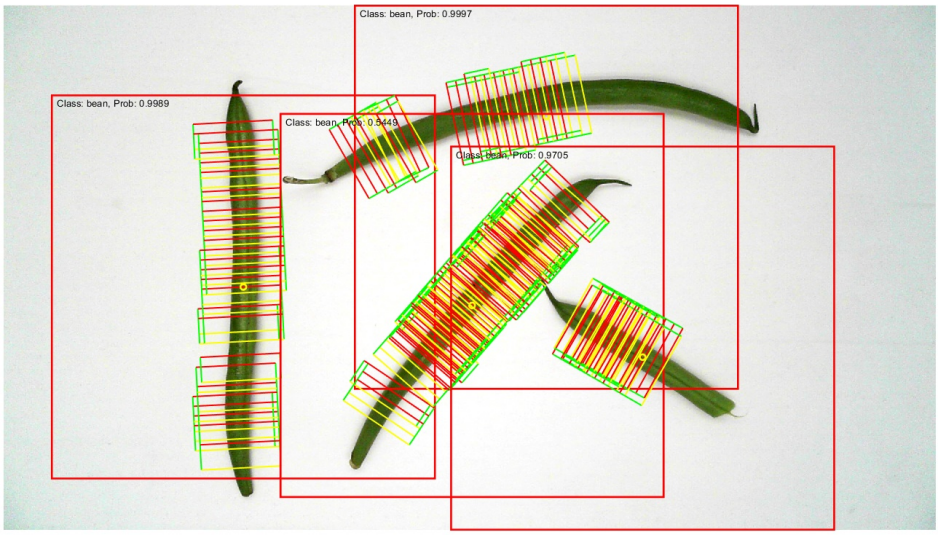
Selecting for the proposed metrics improved OS and OE by 10.3% and 5.1%, respectively, relative to the random selection baseline. In terms of grasp rate, this corresponds to an improvement of 6.0% for known objects and 5.3% for novel objects. Compared to the highest softmax baseline—a selection policy similar to the methods employed by other works—OS and OE were improved by 6.8% and 1.9%, respectively, and grasp rates were improved by 5.3% for known objects and 2.7% for novel objects. In addition, selecting for grasp quality metrics OS and OE—as opposed to selecting for grasp rate—showed improvement in the resultant grasp quality and grasp rate. Optimising for the proposed metrics improved OS and OE by 2.5% and 2.7%, respectively, compared to optimising for the probability of a successful grasp. Grasp rates were improved by 4.6% for known objects and 2.5% for novel objects. Grasps selected via the proposed methodology were qualitatively improved in terms of stability and general suitability, yielding better manipulation quality.

The research presented in this thesis aims to build toward a robust and flexible autonomous handling procedure that grasps objects such that may be manipulated accurately. This work posits that such a system is key for industry application, needed to further close the gap between manual and fully automated production.

11.2 Future direction of this research

The 3-stage method promotes integration and manipulator-agnostic operation, in that component blocks may be swapped to accommodate alternate functionality, hardware and information sources available to the system. In this research, the proposed methodology was implemented on a prototype system for physical validation purposes, resulting in a real-world demonstration of the utility of the 3-stage process. Consequently, the resulting stage 2, grasp detection network is strictly associated with the small, 2-fingered gripper attachment offered by the Dobot Magician system. The goal of this stage is candidate generation, with the softmax output $K_{I_{RGW}}$ serving as one of many inputs to the following stage. Therefore, custom networks that correspond to a new gripper size and design may be substituted—enabling straightforward implementation of the proposed method across multiple systems and various applications. Alternatively, scale-invariant learning approaches could be considered to mitigate the need for proprietary networks when integrating with new hardware. However, universal methods tend to lack precision. As such, it may be more appropriate to train new learning components with the new hardware. Since hardware is often standardised throughout industry, unique datasets and networks may be established to enable plug-and-play functionality.

To briefly explore the flexibility of the 3-stage methodology, stage 1 and stage 2 networks were interchanged for a limited case study related to horticultural sorting. As discussed in Chapter 10.3, the prototype load-cell system only allows for the COG computation of single objects in isolation. However, this issue can be largely mitigated through vision-based approximations. To enable multi-object application, the COG position $[x_{t_{COG}}, y_{t_{COG}}]$ was substituted for the object centroid position $[x_o, y_o]$. Some images related to this study are illustrated in Figure 11.1.



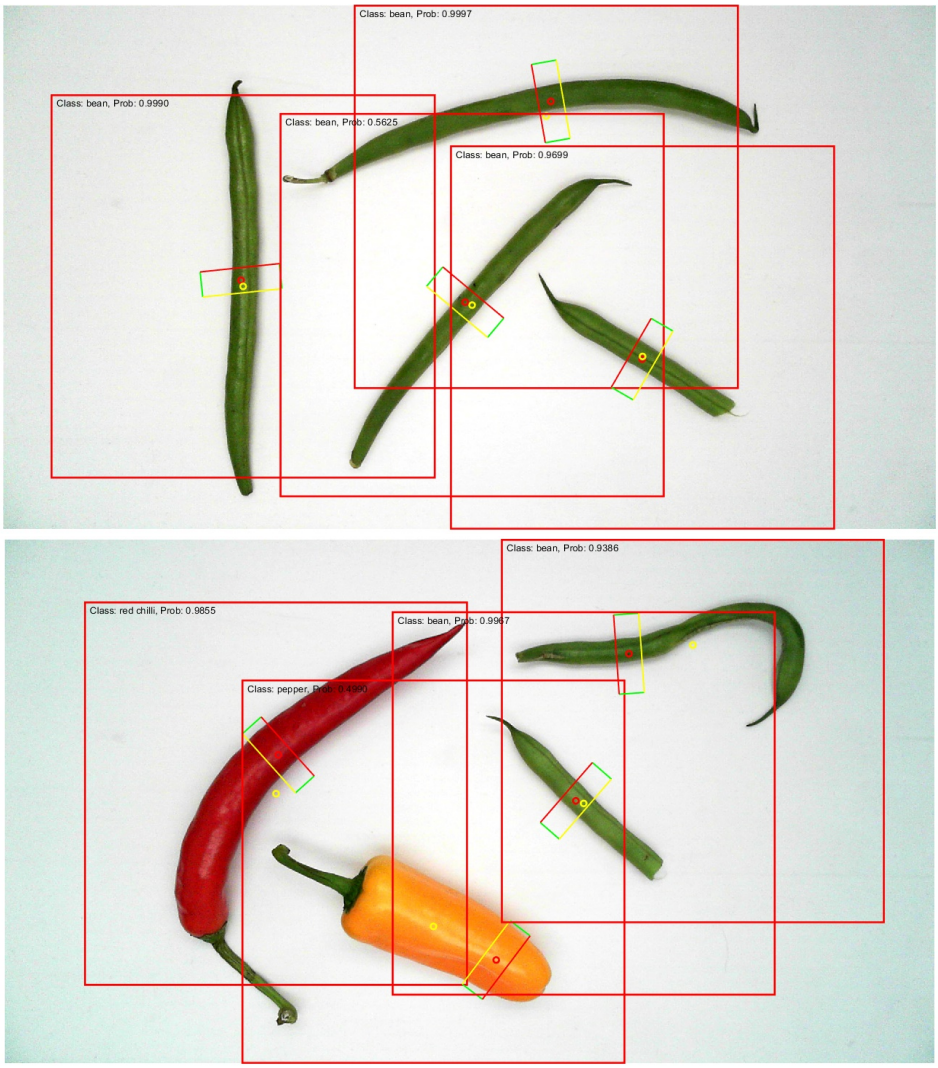


***Figure 11.1*. Various images related to a horticultural sorting application of the proposed methodology. The yellow circle represents the COG-vision approximation of an object and the red circle represents the centre of the grasp.**

The output from the stage 1 component was not used during the grasp generation or selection policy in this study. The future goal of this component relates to a large database of successful grasps, where unique objects are coupled with their grasp outcome. By identifying an object which has been previously well-handled, the grasp detection and selection stages may be avoided by fitting an established grasp configuration to the known object directly. Moreover, the functionality of this component extends to object profiling. Thus, stage 3 selection can be improved by associating information with a specific object profile, e.g., object mass, deformability, fragility, special contact or care instructions, surface properties, etc.

As noted previously, Matlab is considered an excellent platform for research and development. The tools provided by this software are relatively simple use, while still providing a range of functionality. This allows for faster prototype times and a more palatable programming environment. However, Matlab is not usually employed in the final iteration of a project, nor is Matlab common throughout industry. Python on the other hand, is an ubiquitous, general-purpose programming language. Moreover, Python implementation is expected to provide a significant computational boost and simplify

communications with hardware. Future work should consider migrating to Python and consequently, TensorFlow.

As discussed in Chapter 10.2, the rectangular grasping window representation I_{RGW} used in this work does not capture the physical dimensions of the gripper outside of this window. Therefore, no wider context is available when generating or selecting grasps. This becomes problematic in fringe cases where the rectangle itself may be suitable, but the surrounding area is not (Figure 11.2—a, b, c). This issue was explored briefly by Pinto and Gupta [60]. They included a rotation-variant region of interest around their grasp sample as input to their candidate generation component—essentially providing additional context. Future works should explore a similar representation. Figure 11.2—d illustrates a potential rotation-invariant mode of implementation for such a context window and the resulting utility thereof.

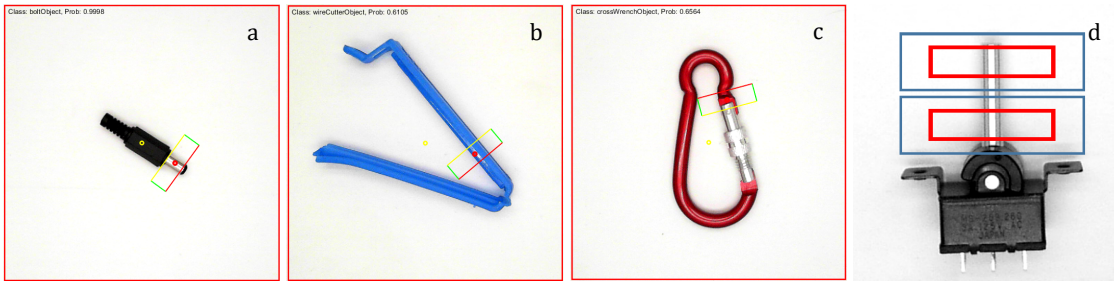


Figure 11.2. **(a), (b), (c)—various grasps demonstrating the important of context. Implementing such grasps may result in object-gripper collisions. (d)—illustration of candidate grasping windows and the associated rotation-invariant context windows proposed for future works. Traditional grasping rectangles are shown in red. Blue rectangles represent a context window. Notice the additional context of the bottom grasp, which is close to collision—compared to the top grasp.**

With steady advances in GPU technology (Chapter 2.1), single-pass networks have gained significant popularity. Consequently, many works have utilised this type of network in this field of research [58, 69, 72, 152, 159]. In this work, a single-pass network would allow for the prediction of grasp pose $pose_t\{x_t, y_t, \theta_t, h_t, w_t\}$ from input image I_t directly. This method may also avoid issues with context. Note that high-end GPU resources may be required to facilitate this approach, compared to the low-end GPU used in this study.

Implementation of the proposed metrics is relatively simple, requiring a single top-down camera. By measuring the error introduced by the grasp trial itself, OS and OE can be used to assess grasp outcome, quantify grasp quality and provide useful criteria for network training. Therefore, the proposed measures may be of interest to the wider community. To promote further development and community involvement, many project-related resources such as Matlab source code, datasets, networks and microcontroller code will be made available at: https://drive.google.com/open?id=1VsEjCl6hrX3FeL9VRF-J9CzVL7JHXO15.

Reference List

[1] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic grasping of novel objects using vision," *The International Journal of Robotics Research,* vol. 27, no. 2, pp. 157-173, 2008.

[2] A. Ramisa, G. Alenya, F. Moreno-Noguer, and C. Torras, "Using depth and appearance features for informed robot grasping of highly wrinkled clothes," in *2012 IEEE International Conference on Robotics and Automation*, 2012: IEEE, pp. 1703-1708.

[3] D. Seita *et al.*, "Robot bed-making: Deep transfer learning using depth sensing of deformable fabric," *arXiv preprint arXiv:1809.09810,* 2018.

[4] M. Beetz *et al.*, "Robotic roommates making pancakes," in *2011 11th IEEE-RAS International Conference on Humanoid Robots*, 2011: IEEE, pp. 529-536.

[5] Moley Robotics, "The Moley Robotic Kitchen - Mission & Goals," ed, 2015.

[6] C. Militaru, A. D. Mezei, and L. Tamas, "Object handling in cluttered indoor environment with a mobile manipulator," in *2016 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, 19-21 May 2016 2016, pp. 1-6, doi: 10.1109/AQTR.2016.7501382.

[7] K. Wu, R. Ranasinghe, and G. Dissanayake, "Active recognition and pose estimation of household objects in clutter," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015: IEEE, pp. 4230-4237.

[8] W. Miyazaki and J. Miura, "Object placement estimation with occlusions and planning of robotic handling strategies," in *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, 3-7 July 2017 2017, pp. 602-607, doi: 10.1109/AIM.2017.8014083.

[9] S. Levine, N. Wagener, and P. Abbeel, "Learning contact-rich manipulation skills with guided policy search," *arXiv preprint arXiv:1501.05611,* 2015.

[10] M. Gualtieri, A. t. Pas, K. Saenko, and R. Platt, "High precision grasp pose detection in dense clutter," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 9-14 Oct. 2016 2016, pp. 598-605, doi: 10.1109/IROS.2016.7759114.

[11] M. Gualtieri, A. t. Pas, and R. Platt, "Pick and Place Without Geometric Object Models," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 21-25 May 2018 2018, pp. 7433-7440, doi: 10.1109/ICRA.2018.8460553.

[12] L. Manuelli, W. Gao, P. Florence, and R. Tedrake, "kPAM: KeyPoint Affordances for Category-Level Robotic Manipulation," *arXiv preprint arXiv:1903.06684,* 2019.

[13] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *The International Journal of Robotics Research,* vol. 34, no. 4-5, pp. 705-724, 2015.

[14] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, "Deep object pose estimation for semantic robotic grasping of household objects," *arXiv preprint arXiv:1809.10790,* 2018.

[15] Amazon Robotics LLC. "2017 Amazon Robotics Challenge Official Rules." https://www.amazonrobotics.com/site/binaries/content/assets/amazonrobotics/arc/2017-amazon-robotics-challenge-rules-v3.pdf (accessed 24 June, 2019).

[16] D. Morrison *et al.*, "Cartman: The low-cost cartesian manipulator that won the amazon robotics challenge," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018: IEEE, pp. 7757-7764.

[17] M. Schwarz *et al.*, "Fast Object Learning and Dual-arm Coordination for Cluttered Stowing, Picking, and Packing," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 21-25 May 2018 2018, pp. 3347-3354, doi: 10.1109/ICRA.2018.8461195.

[18] A. Zeng *et al.*, "Multi-view self-supervised deep learning for 6D pose estimation in the Amazon Picking Challenge," in *2017 IEEE International Conference on Robotics*

and Automation (ICRA), 29 May-3 June 2017 2017, pp. 1386-1383, doi: 10.1109/ICRA.2017.7989165.
[19] M. Schwarz *et al.*, "NimbRo Picking: Versatile part handling for warehouse automation," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017: IEEE, pp. 3032-3039.
[20] A. Zeng *et al.*, "Robotic Pick-and-Place of Novel Objects in Clutter with Multi-Affordance Grasping and Cross-Domain Image Matching," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 21-25 May 2018 2018, pp. 1-8, doi: 10.1109/ICRA.2018.8461044.
[21] C. Hernandez *et al.*, "Team delft's robot winner of the amazon picking challenge 2016," in *Robot World Cup*, 2016: Springer, pp. 613-624.
[22] R. Jonschkowski, C. Eppner, S. Höfer, R. Martín-Martín, and O. Brock, "Probabilistic multi-class segmentation for the amazon picking challenge," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016: IEEE, pp. 1-7.
[23] C. Zhihong, Z. Hebin, W. Yanbo, L. Binyan, and L. Yu, "A vision-based robotic grasping system using deep learning for garbage sorting," in *2017 36th Chinese Control Conference (CCC)*, 26-28 July 2017 2017, pp. 11223-11226, doi: 10.23919/ChiCC.2017.8029147.
[24] E. Klingbeil, D. Rao, B. Carpenter, V. Ganapathi, A. Y. Ng, and O. Khatib, "Grasping with application to an autonomous checkout robot," in *2011 IEEE International Conference on Robotics and Automation*, 9-13 May 2011 2011, pp. 2837-2844, doi: 10.1109/ICRA.2011.5980287.
[25] Y. Zhang, H. Qiao, J. Su, K. Huang, and T. Fukuda, "A vison-based grasping strategy for the mineral sorting," in *Proceedings of the 10th World Congress on Intelligent Control and Automation*, 2012: IEEE, pp. 3454-3459.
[26] Z. Cenev, J. Venäläinen, V. Sariola, and Q. Zhou, "Object tracking in robotic micromanipulation by supervised ensemble learning classifier," in *2016 International Conference on Manipulation, Automation and Robotics at Small Scales (MARSS)*, 2016: IEEE, pp. 1-5.
[27] A. Peñalver, J. J. Fernández, and P. J. Sanz, "Autonomous underwater grasping using multi-view laser reconstruction," in *OCEANS 2017-Aberdeen*, 2017: IEEE, pp. 1-5.
[28] J. Shi and G. S. Koonjul, "Real-time grasping planning for robotic bin-picking and kitting applications," *IEEE Transactions on Automation Science and Engineering,* vol. 14, no. 2, pp. 809-819, 2017.
[29] H. Cheng, H. Chen, and Y. Liu, "Object handling using Autonomous Industrial Mobile Manipulator," in *2013 IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems*, 26-29 May 2013 2013, pp. 36-41, doi: 10.1109/CYBER.2013.6705416.
[30] M. Ferrati, S. Nardi, A. Settimi, H. Marino, and L. Pallottino, "Multi-object handling for robotic manufacturing," in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, 23-26 Oct. 2016 2016, pp. 6887-6893, doi: 10.1109/IECON.2016.7793936.
[31] Y. Angal and A. Gade, "LabVIEW controlled robot for object handling using NI myRIO," in *2016 IEEE International Conference on Advances in Electronics, Communication and Computer Technology (ICAECCT)*, 2-3 Dec. 2016 2016, pp. 167-171, doi: 10.1109/ICAECCT.2016.7942576.
[32] S. Anton and F. D. Anton, "Recognizing and handling articulated objects in robotic tasks," in *2017 40th International Conference on Telecommunications and Signal Processing (TSP)*, 5-7 July 2017 2017, pp. 666-669, doi: 10.1109/TSP.2017.8076070.

[33] Y. Bekiroglu, J. Laaksonen, J. A. Jorgensen, V. Kyrki, and D. Kragic, "Assessing grasp stability based on learning and haptic data," *IEEE Transactions on Robotics,* vol. 27, no. 3, pp. 616-629, 2011.
[34] A. M. Welhenge, R. D. Wijesinghe, and R. M. T. P. Rajakaruna, "Robotic gripper design to handle an arbitrarily shaped object by emulating human finger motion," in *2015 8th International Conference on Ubi-Media Computing (UMEDIA)*, 24-26 Aug. 2015 2015, pp. 330-334, doi: 10.1109/UMEDIA.2015.7297480.
[35] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International Journal of Robotics Research,* vol. 37, no. 4-5, pp. 421-436, 2018.
[36] D. Kalashnikov *et al.*, "Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation," *arXiv preprint arXiv:1806.10293,* 2018.
[37] RobotWorx, "Pick and Place Robots," ed, 2011.
[38] OMRON Asia Pacific, "Omron Integrated Robotic Automation Solution," ed, 2017.
[39] ABB Group. (2013) The future is now. *International customer magazine from ABB Robotics*. Available: https://library.e.abb.com/public/b9d7a5043edb6a87c1257c3d004a0248/ABB_Robotics_2_2013_online.pdf
[40] J. Mahler *et al.*, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," *arXiv preprint arXiv:1703.09312,* 2017.
[41] A. ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *The International Journal of Robotics Research,* vol. 36, no. 13-14, pp. 1455-1473, 2017.
[42] E. Johns, S. Leutenegger, and A. J. Davison, "Deep learning a grasp function for grasping under gripper pose uncertainty," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 9-14 Oct. 2016 2016, pp. 4461-4468, doi: 10.1109/IROS.2016.7759657.
[43] J. Kim, K. Iwamoto, J. J. Kuffner, Y. Ota, and N. S. Pollard, "Physically Based Grasp Quality Evaluation Under Pose Uncertainty," *IEEE Transactions on Robotics,* vol. 29, no. 6, pp. 1424-1439, 2013, doi: 10.1109/TRO.2013.2273846.
[44] L. U. Odhner, R. R. Ma, and A. M. Dollar, "Open-Loop Precision Grasping With Underactuated Hands Inspired by a Human Manipulation Strategy," *IEEE Transactions on Automation Science and Engineering,* vol. 10, no. 3, pp. 625-633, 2013, doi: 10.1109/TASE.2013.2240298.
[45] L. Wang, J. DelPreto, S. Bhattacharyya, J. Weisz, and P. K. Allen, "A highly-underactuated robotic hand with force and joint angle sensors," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011: IEEE, pp. 1380-1385.
[46] I. Sainul, S. Deb, and A. Deb, "A novel object slicing based grasp planner for 3D object grasping using underactuated robot gripper," *arXiv preprint arXiv:1907.09142,* 2019.
[47] S. Yao, Q. Zhan, M. Ceccarelli, G. Carbone, and Z. Lu, "Analysis and grasp strategy modeling for underactuated multi-fingered robot hand," in *2009 International Conference on Mechatronics and Automation*, 2009: IEEE, pp. 2817-2822.
[48] T. Laliberté, L. Birglen, and C. Gosselin, "Underactuation in robotic grasping hands," *Machine Intelligence & Robotic Control,* vol. 4, no. 3, pp. 1-11, 2002.
[49] R. Shauri, N. Salleh, and A. Hadi, "PID position control of 7-DOF three-fingered robotic hand for grasping task," in *2014 IEEE International Conference on Control System, Computing and Engineering (ICCSCE 2014)*, 2014: IEEE, pp. 70-74.
[50] A. M. Dollar and R. D. Howe, "Simple, reliable robotic grasping for human environments," in *2008 IEEE International Conference on Technologies for Practical Robot Applications*, 2008: IEEE, pp. 156-161.

[51] A. M. Dollar and R. D. Howe, "Simple, robust autonomous grasping in unstructured environments," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, 2007: IEEE, pp. 4693-4700.
[52] A. M. Dollar and R. D. Howe, "The highly adaptive SDM hand: Design and performance evaluation," *The international journal of robotics research,* vol. 29, no. 5, pp. 585-597, 2010.
[53] A. Zelenak, C. Brabec, J. Thompson, J. Hashem, B. Fernandez, and M. Pryor, "Intelligent Grasping with the Robotic Opposable Thumb," *Applied Artificial Intelligence,* vol. 28, no. 8, pp. 737-750, 2014.
[54] F. Rothling, R. Haschke, J. J. Steil, and H. Ritter, "Platform portable anthropomorphic grasping with the bielefeld 20-dof shadow and 9-dof tum hand," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007: IEEE, pp. 2951-2956.
[55] E. Brown *et al.*, "Universal robotic gripper based on the jamming of granular material," *Proceedings of the National Academy of Sciences,* vol. 107, no. 44, pp. 18809-18814, 2010.
[56] K. Harada, K. Nagata, T. Tsuji, N. Yamanobe, A. Nakamura, and Y. Kawai, "Probabilistic approach for object bin picking approximated by cylinders," in *2013 IEEE International Conference on Robotics and Automation*, 2013: IEEE, pp. 3742-3747.
[57] V. Kumar, Q. Wang, W. Minghua, S. Rizwan, S. Shaikh, and X. Liu, "Computer vision based object grasping 6DoF robotic arm using picamera," in *2018 4th International Conference on Control, Automation and Robotics (ICCAR)*, 2018: IEEE, pp. 111-115.
[58] S. Kumra and C. Kanan, "Robotic grasp detection using deep convolutional neural networks," *arXiv preprint arXiv:1611.08036,* 2017.
[59] A. Saxena, J. Driemeyer, J. Kearns, and A. Y. Ng, "Robotic grasping of novel objects," in *Advances in neural information processing systems*, 2007, pp. 1209-1216.
[60] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50K tries and 700 robot hours," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 16-21 May 2016 2016, pp. 3406-3413, doi: 10.1109/ICRA.2016.7487517.
[61] L. Pinto, J. Davidson, and A. Gupta, "Supervision via competition: Robot adversaries for learning tasks," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017: IEEE, pp. 1601-1608.
[62] A. t. Pas and R. Platt, "Using geometry to detect grasps in 3d point clouds," *arXiv preprint arXiv:1501.03100,* 2015.
[63] A. Sahbani, S. El-Khoury, and P. Bidaud, "An overview of 3D object grasp synthesis algorithms," *Robotics and Autonomous Systems,* vol. 60, no. 3, pp. 326-336, 2012.
[64] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-Driven Grasp Synthesis—A Survey," *IEEE Transactions on Robotics,* vol. 30, no. 2, pp. 289-309, 2014, doi: 10.1109/TRO.2013.2289018.
[65] Y. Jiang, S. Moseson, and A. Saxena, "Efficient grasping from rgbd images: Learning using a new rectangle representation," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011: IEEE, pp. 3304-3311.
[66] C. Sun, Y. Yu, H. Liu, and J. Gu, "Robotic grasp detection using extreme learning machine," in *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 6-9 Dec. 2015 2015, pp. 1115-1120, doi: 10.1109/ROBIO.2015.7418921.
[67] Cornell University. "Robot Learning Lab: Personal Robotics, Co-Robots, Robotic Perception. Computer Science Department, Cornell University. ." Cornell University. http://pr.cs.cornell.edu/grasping/rect_data/data.php (accessed 19 February, 2020).
[68] H. Cheng and M. Q. Meng, "A Grasp Pose Detection Scheme with an End-to-End CNN Regression Approach," in *2018 IEEE International Conference on Robotics and*

Biomimetics (ROBIO), 12-15 Dec. 2018 2018, pp. 544-549, doi: 10.1109/ROBIO.2018.8665219.

[69] J. Redmon and A. Angelova, "Real-time grasp detection using convolutional neural networks," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, 2015: IEEE, pp. 1316-1322.

[70] F. Chu, R. Xu, and P. A. Vela, "Real-World Multiobject, Multigrasp Detection," *IEEE Robotics and Automation Letters,* vol. 3, no. 4, pp. 3355-3362, 2018, doi: 10.1109/LRA.2018.2852777.

[71] Z. Wang, Z. Li, B. Wang, and H. Liu, "Robot grasp detection using multimodal deep convolutional neural networks," *Advances in Mechanical Engineering,* vol. 8, no. 9, p. 1687814016668077, 2016.

[72] D. Morrison, P. Corke, and J. Leitner, "Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach," *arXiv preprint arXiv:1804.05172,* 2018.

[73] S. Caldera, A. Rassau, and D. Chai, "Robotic Grasp Pose Detection Using Deep Learning," in *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 18-21 Nov. 2018 2018, pp. 1966-1972, doi: 10.1109/ICARCV.2018.8581091.

[74] L. Pusong, B. DeRose, J. Mahler, J. Ojea, A. Tanwani, and K. Golberg, "Dex-net as a service (dnaas): A cloud-based robust robot grasp planning system," in *14th International Conference on Automation Science and Engineering (CASE)*, 2018, pp. 1-8.

[75] U. Viereck, A. t. Pas, K. Saenko, and R. Platt, "Learning a visuomotor controller for real world robotic grasping using simulated depth images," *arXiv preprint arXiv:1706.04652,* 2017.

[76] A. Zeng, S. Song, S. Welker, J. Lee, A. Rodriguez, and T. Funkhouser, "Learning synergies between pushing and grasping with self-supervised deep reinforcement learning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018: IEEE, pp. 4238-4245.

[77] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *The Journal of Machine Learning Research,* vol. 17, no. 1, pp. 1334-1373, 2016.

[78] E. Arruda, J. Wyatt, and M. Kopicki, "Active vision for dexterous grasping of novel objects," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 9-14 Oct. 2016 2016, pp. 2881-2888, doi: 10.1109/IROS.2016.7759446.

[79] A. M. Dollar and R. D. Howe, "Joint coupling design of underactuated hands for unstructured environments," *The International Journal of Robotics Research,* vol. 30, no. 9, pp. 1157-1169, 2011.

[80] Z. Ding, N. Paperno, K. Prakash, and A. Behal, "An Adaptive Control-Based Approach for 1-Click Gripping of Novel Objects Using a Robotic Manipulator," *IEEE Transactions on Control Systems Technology,* pp. 1-8, 2018, doi: 10.1109/TCST.2018.2821651.

[81] M. Kopicki, R. Detry, F. Schmidt, C. Borst, R. Stolkin, and J. L. Wyatt, "Learning dexterous grasps that generalise to novel objects by combining hand and contact models," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 31 2014-June 7 2014 2014, pp. 5358-5365, doi: 10.1109/ICRA.2014.6907647.

[82] A. K. Goins, R. Carpenter, W.-K. Wong, and R. Balasubramanian, "Evaluating the efficacy of grasp metrics for utilization in a gaussian process-based grasp predictor," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014: IEEE, pp. 3353-3360.

[83] A. Morales, E. Chinellato, A. H. Fagg, and A. P. Del Pobil, "Using experience for assessing grasp reliability," *International Journal of Humanoid Robotics,* vol. 1, no. 04, pp. 671-691, 2004.

[84] M. Gualtieri and R. Platt, "Viewpoint selection for grasp detection," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017: IEEE, pp. 258-264.
[85] J. Varley, J. Weisz, J. Weiss, and P. Allen, "Generating multi-fingered robotic grasps via deep learning," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 28 Sept.-2 Oct. 2015 2015, pp. 4415-4420, doi: 10.1109/IROS.2015.7354004.
[86] A. Saxena, L. L. Wong, and A. Y. Ng, "Learning grasp strategies with partial shape information," in *AAAI*, 2008, vol. 3, no. 2, pp. 1491-1494.
[87] Q. V. Le, D. Kamm, A. F. Kara, and A. Y. Ng, "Learning to grasp objects with multiple contact points," in *2010 IEEE International Conference on Robotics and Automation*, 3-7 May 2010 2010, pp. 5062-5069, doi: 10.1109/ROBOT.2010.5509508.
[88] F. Lévesque, B. Sauvet, P. Cardou, and C. Gosselin, "A model-based scooping grasp for the autonomous picking of unknown objects with a two-fingered gripper," *Robotics and Autonomous Systems,* vol. 106, pp. 14-25, 2018/08/01/ 2018, doi: https://doi.org/10.1016/j.robot.2018.04.003.
[89] S. James *et al.*, "Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12627-12637.
[90] D. Rodriguez, C. Cogswell, S. Koo, and S. Behnke, "Transferring grasping skills to novel instances by latent space non-rigid registration," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018: IEEE, pp. 1-8.
[91] IEEE Robotics & Automation Magazine, "Special Issue on Replicable and Measurable Robotics Research," *IEEE Robotics & Automation Magazine,* vol. 21, no. 3, pp. 153-153, 2014, doi: 10.1109/MRA.2014.2346083.
[92] F. Bonsignorio and A. P. Del Pobil, "Toward replicable and measurable robotics research [from the guest editors]," *IEEE Robotics & Automation Magazine,* vol. 22, no. 3, pp. 32-35, 2015.
[93] J. Mahler *et al.*, "Guest editorial open discussion of robot grasping benchmarks, protocols, and metrics," *IEEE Transactions on Automation Science and Engineering,* vol. 15, no. 4, pp. 1440-1442, 2018.
[94] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The YCB object and Model set: Towards common benchmarks for manipulation research," in *2015 International Conference on Advanced Robotics (ICAR)*, 27-31 July 2015 2015, pp. 510-517, doi: 10.1109/ICAR.2015.7251504.
[95] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in manipulation research: The YCB object and model set and benchmarking protocols," *arXiv preprint arXiv:1502.03143,* 2015.
[96] J. Leitner *et al.*, "The ACRV picking benchmark: A robotic shelf picking benchmark to foster reproducible research," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017: IEEE, pp. 4705-4712.
[97] J. Leitner. "The ARCV Picking Benchmark (APB)." http://juxi.net/acrv-picking-benchmark/ (accessed 9 July, 2019).
[98] C. Rubert, B. León, A. Morales, and J. Sancho-Bru, "Characterisation of grasp quality metrics," *Journal of Intelligent & Robotic Systems,* vol. 89, no. 3-4, pp. 319-342, 2018.
[99] S. Ulbrich *et al.*, "The OpenGRASP benchmarking suite: An environment for the comparative analysis of grasping and dexterous manipulation," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011: IEEE, pp. 1761-1767.
[100] A. T. Miller and P. K. Allen, "Graspit! a versatile simulator for robotic grasping," *IEEE Robotics & Automation Magazine,* vol. 11, no. 4, pp. 110-122, 2004.

[101] G. Kootstra, M. Popović, J. A. Jørgensen, D. Kragic, H. G. Petersen, and N. Krüger, "VisGraB: A benchmark for vision-based grasping," *Paladyn, Journal of Behavioral Robotics,* vol. 3, no. 2, pp. 54-62, 2012.
[102] Y. Xiang, R. Mottaghi, and S. Savarese, "Beyond PASCAL: A benchmark for 3D object detection in the wild," in *IEEE Winter Conference on Applications of Computer Vision*, 24-26 March 2014 2014, pp. 75-82, doi: 10.1109/WACV.2014.6836101.
[103] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision,* vol. 88, no. 2, pp. 303-338, 2010.
[104] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097-1105.
[105] T.-Y. Lin *et al.*, "Microsoft coco: Common objects in context," in *European conference on computer vision*, 2014: Springer, pp. 740-755.
[106] J. Watson, J. Hughes, and F. Iida, "Real-world, real-time robotic grasping with convolutional neural networks," in *Annual Conference Towards Autonomous Robotic Systems*, 2017: Springer, pp. 617-626.
[107] C. Rubert, D. Kappler, A. Morales, S. Schaal, and J. Bohg, "On the relevance of grasp metrics for predicting grasp success," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 24-28 Sept. 2017 2017, pp. 265-272, doi: 10.1109/IROS.2017.8202167.
[108] A. Grau, M. Indri, L. L. Bello, and T. Sauter, "Industrial robotics in factory automation: From the early stage to the Internet of Things," in *IECON 2017-43rd Annual Conference of the IEEE Industrial Electronics Society*, 2017: IEEE, pp. 6159-6164.
[109] International Federation of Robotics, "Executive Summary: World Robotics 2019 Industrial Robots," 2019. [Online]. Available: https://ifr.org/downloads/press2018/Executive%20Summary%20WR%202019%20Industrial%20Robots.pdf
[110] International Federation of Robotics, "IFR Press Conference," in *IFR Press Conference*, ed. Shanghai, 2019.
[111] S. Vaidya, P. Ambad, and S. Bhosle, "Industry 4.0–a glimpse," *Procedia Manufacturing,* vol. 20, pp. 233-238, 2018.
[112] J. Trojanowska, K. Żywicki, M. L. R. Varela, and J. M. Machado, "Shortening changeover time - An industrial study," in *2015 10th Iberian Conference on Information Systems and Technologies (CISTI)*, 17-20 June 2015 2015, pp. 1-6, doi: 10.1109/CISTI.2015.7170373.
[113] N. Keddis, G. Kainz, C. Buckl, and A. Knoll, "Towards adaptable manufacturing systems," in *2013 IEEE International Conference on Industrial Technology (ICIT)*, 25-28 Feb. 2013 2013, pp. 1410-1415, doi: 10.1109/ICIT.2013.6505878.
[114] H. ElMaraghy and W. ElMaraghy, "Smart adaptable assembly systems," *Procedia CIRP,* vol. 44, pp. 4-13, 2016.
[115] Y. Koren, X. Gu, and W. Guo, "Reconfigurable manufacturing systems: Principles, design, and future trends," *Frontiers of Mechanical Engineering,* vol. 13, no. 2, pp. 121-136, 2018.
[116] N. V. K. Jasti and R. Kodali, "Lean production: literature review and trends," *International Journal of Production Research,* vol. 53, no. 3, pp. 867-885, 2015.
[117] J. Varsaluoma *et al.*, "Usage Data Analytics for Human-Machine Interactions with Flexible Manufacturing Systems: Opportunities and Challenges," in *2017 21st International Conference Information Visualisation (IV)*, 11-14 July 2017 2017, pp. 427-434, doi: 10.1109/iV.2017.38.
[118] K. Al-Gumaei, A. Müller, J. N. Weskamp, C. Santo Longo, F. Pethig, and S. Windmann, "Scalable Analytics Platform for Machine Learning in Smart Production

Systems," in *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2019: IEEE, pp. 1155-1162.
[119] M. Dotoli, A. Fay, M. Miśkowicz, and C. Seatzu, "An overview of current technologies and emerging trends in factory automation," *International Journal of Production Research,* pp. 1-21, 2018.
[120] R. McLean, A. J. Walker, and G. Bright, "An artificial neural network driven decision-making system for manufacturing disturbance mitigation in reconfigurable systems," in *2017 13th IEEE International Conference on Control & Automation (ICCA)*, 3-6 July 2017 2017, pp. 695-700, doi: 10.1109/ICCA.2017.8003144.
[121] M. Hedelind and S. Kock, "Requirements on flexible robot systems for small parts assembly: A case study," in *2011 IEEE International Symposium on Assembly and Manufacturing (ISAM)*, 25-27 May 2011 2011, pp. 1-7, doi: 10.1109/ISAM.2011.5942356.
[122] S. Kock *et al.*, "Robot concept for scalable, flexible assembly automation: A technology study on a harmless dual-armed robot," in *2011 IEEE International Symposium on Assembly and Manufacturing (ISAM)*, 25-27 May 2011 2011, pp. 1-5, doi: 10.1109/ISAM.2011.5942358.
[123] J. Woetzel *et al.*, "China and the world: Inside the dynamics of a changing relationship," McKinsey Global Institute, 2019.
[124] Information Technology & Innovation Foundation. "Robotics and the Future of Production and Work." https://itif.org/publications/2019/10/15/robotics-and-future-production-and-work (accessed 20 November, 2019).
[125] International Federation of Robotics. "India's Robot Wonder." https://ifr.org/ifr-press-releases/news/indias-robot-wonder (accessed 21 November, 2019).
[126] Fanuc America Corporation. "Robotic Solutions for Assembly - Industries primed for fast, small parts assembly processes." https://www.fanucamerica.com/industrial-solutions/manufacturing-applications/assembly-robots (accessed 24 August, 2019).
[127] Mitsubishi Electric Automation Incorporated, "Mitsubishi Electric Robot Pick and Place - Thermal Relay," ed, 2014.
[128] Kawasaki Robotics, "Assembling Electronic Devices - Kawasaki MC004N robot," ed, 2014.
[129] Omron Robotics and Safety Technologies Incorporated, "Adept Cobra - Brush Assembly Application," ed, 2015.
[130] Denso Robotics, "Small Assembly Robots: Comparing the Cost of Ownership of Different Brands." [Online]. Available: https://api.densorobotics.com/downloads/file/3
[131] Farason Corporation. "Robotic Parts Unscrambling & Loading System." http://www.farason.com/wordpress1/home/welcome-to-farason/solutions/by-industry/pharmaceutical/robotic-part-unscrambling-loading-system/ (accessed 24 August, 2019).
[132] T. M. Anandan. "Robot–Scientist Collaboration (and Separation) in Lab Automation." https://www.robotics.org/content-detail.cfm/Industrial-Robotics-Industry-Insights/Robot-Scientist-Collaboration-and-Separation-in-Lab-Automation/content_id/5445 (accessed 24 August, 2019).
[133] J. McCarthy, M. L. Minsky, N. Rochester, and C. E. Shannon, "A proposal for the Dartmouth summer research project on artificial intelligence," 1955. [Online]. Available: http://www-formal.stanford.edu/jmc/history/dartmouth/dartmouth.html
[134] K. C. Morris, C. Schlenoff, and V. Srinivasan, "Guest Editorial A Remarkable Resurgence of Artificial Intelligence and Its Impact on Automation and Autonomy," *IEEE Transactions on Automation Science and Engineering,* vol. 14, no. 2, pp. 407-409, 2017, doi: 10.1109/TASE.2016.2640778.

[135] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016.
[136] J. Best. "IBM Watson: The inside story of how the Jeopardy-winning supercomputer was born, and what it wants to do next." https://www.techrepublic.com/article/ibm-watson-the-inside-story-of-how-the-jeopardy-winning-supercomputer-was-born-and-what-it-wants-to-do-next/ (accessed 3 November, 2019).
[137] B. Upbin. "IBM's Watson Gets Its First Piece Of Business In Healthcare." https://www.forbes.com/sites/bruceupbin/2013/02/08/ibms-watson-gets-its-first-piece-of-business-in-healthcare/#7b184ce65402 (accessed 2 November, 2019).
[138] IBM Corporation, "IBM Watson Health in Oncology: Scientific Evidence 2019," 2019. [Online]. Available: https://www.ibm.com/downloads/cas/0ZRYPWL9
[139] S. R. Granter, A. H. Beck, and D. J. Papke Jr, "AlphaGo, deep learning, and the future of the human microscopist," *Archives of pathology & laboratory medicine,* vol. 141, no. 5, pp. 619-621, 2017.
[140] D. Silver and D. Hassabis, "AlphaGo Zero: Starting from scratch," ed, 2017.
[141] S. Bazrafkan, T. Nedelcu, P. Filipczuk, and P. Corcoran, "Deep learning for facial expression recognition: A step closer to a smartphone that knows your moods," in *Consumer Electronics (ICCE), 2017 IEEE International Conference on*, 2017: IEEE, pp. 217-220.
[142] M. T. Vega, D. C. Mocanu, J. Famaey, S. Stavrou, and A. Liotta, "Deep learning for quality assessment in live video streaming," *IEEE signal processing letters,* vol. 24, no. 6, pp. 736-740, 2017.
[143] J. Lemley, S. Bazrafkan, and P. Corcoran, "Deep Learning for Consumer Devices and Services: Pushing the limits for machine learning, artificial intelligence, and computer vision," *IEEE Consumer Electronics Magazine,* vol. 6, no. 2, pp. 48-56, 2017.
[144] M. L. Scuri. "Revealing Minerva and addressing toxicity and abusive behaviour in matches." https://blog.faceit.com/revealing-minerva-and-addressing-toxicity-and-abusive-behavior-in-matches-9073914a51c (accessed 28 November, 2019).
[145] A. A. Alurkar *et al.*, "A proposed data science approach for email spam classification using machine learning techniques," in *2017 Internet of Things Business Models, Users, and Networks*, 23-24 Nov. 2017 2017, pp. 1-5, doi: 10.1109/CTTE.2017.8260935.
[146] J. Stilgoe, "Machine learning, social learning and the governance of self-driving cars," *Social studies of science,* vol. 48, no. 1, pp. 25-56, 2018.
[147] M. Bojarski *et al.*, "Explaining how a deep neural network trained with end-to-end learning steers a car," *arXiv preprint arXiv:1704.07911,* 2017.
[148] H. Spens and J. Lindgren, "Using cloud services and machine learning to improve customer support: Study the applicability of the method on voice data," ed, 2018.
[149] T. Lang and M. Rettenmeier, "Understanding consumer behavior with recurrent neural networks," in *Workshop on Machine Learning Methods for Recommender Systems*, 2017.
[150] F. V. Alejandre, N. C. Cortés, and E. A. Anaya, "Feature selection to detect botnets using machine learning algorithms," in *2017 International Conference on Electronics, Communications and Computers (CONIELECOMP)*, 2017: IEEE, pp. 1-7.
[151] S. W. Sidehabi and S. Tandungan, "Statistical and Machine Learning approach in forex prediction based on empirical data," in *2016 International Conference on Computational Intelligence and Cybernetics*, 2016: IEEE, pp. 63-68.
[152] A. Milioto, P. Lottes, and C. Stachniss, "Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in cnns," *arXiv preprint arXiv:1709.06764,* 2017.

[153] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431-3440.
[154] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, 2015: Springer, pp. 234-241.
[155] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence,* vol. 39, no. 12, pp. 2481-2495, 2017.
[156] W. Kuo, A. Angelova, J. Malik, and T.-Y. Lin, "ShapeMask: Learning to Segment Novel Objects by Refining Shape Priors," *arXiv preprint arXiv:1904.03239,* 2019.
[157] N. Kussul, M. Lavreniuk, S. Skakun, and A. Shelestov, "Deep learning classification of land cover and crop types using remote sensing data," *IEEE Geoscience and Remote Sensing Letters,* vol. 14, no. 5, pp. 778-782, 2017.
[158] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," *arXiv preprint,* 2017.
[159] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv preprint arXiv:1804.02767,* 2018.
[160] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "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.
[161] A. El Housseini, A. Toumi, and A. Khenchaf, "Deep Learning for target recognition from SAR images," in *2017 Seminar on Detection Systems Architectures and Technologies (DAT)*, 2017: IEEE, pp. 1-5.
[162] M. Chui, "Artificial intelligence the next digital frontier?," *McKinsey and Company Global Institute,* vol. 47, 2017.
[163] R. Raina, A. Madhavan, and A. Y. Ng, "Large-scale deep unsupervised learning using graphics processors," in *Proceedings of the 26th annual international conference on machine learning*, 2009: ACM, pp. 873-880.
[164] I. Webster. "U.S. Inflation Rate Calculator from 1665 through 2019." https://www.in2013dollars.com/1860-dollars-in-2017?amount=1 (accessed 2 December, 2019).
[165] A. L. Shimpi. "NVIDIA GeForce2 Ultra." https://www.anandtech.com/show/601/15 (accessed 2 December, 2019).
[166] Tech Powerup. "NVIDIA GeForce3 Ti500." https://www.techpowerup.com/gpu-specs/geforce3-ti500.c741 (accessed 2 December, 2019).
[167] M. Witheiler. "NVIDIA GeForce3 Roundup - July 2001." https://www.anandtech.com/show/802 (accessed 2 December, 2019).
[168] M. Witheiler. "NVIDIA GeForce2/3 Titanium Roundup - January 2002." https://www.anandtech.com/show/873 (accessed 2 December, 2019).
[169] M. Hachman. "Update: Nvidia Corp. Announces GeForce4." https://www.extremetech.com/extreme/72964-update-nvidia-corp-announces-geforce4 (accessed 2 December, 2019).
[170] M. Hachman. "Nvidia Launches GeForce 6800 Graphics Family." https://www.extremetech.com/extreme/56092-nvidia-launches-geforce-6800-graphics-family (accessed 2 December, 2019).
[171] D. Wilson. "NVIDIA's GeForce 7800 GTX Hits The Ground Running." https://www.anandtech.com/show/1717 (accessed 2 December, 2019).
[172] P. Mann. "NVIDIA drops prices and announces GeForce 9800 GTX+." https://hexus.net/tech/news/graphics/13912-nvidia-drops-prices-announces-geforce-9800-gtx/ (accessed 2 December, 2019).
[173] B. Lang. "NVIDIA's New GTX 1060 Answers AMD's RX 480, Starts at $249 and Launches July 19th." https://www.roadtovr.com/nvidia-gtx-1060-vr-gpu-price-specs-release-date/ (accessed 2 December, 2019).

[174] B. Thomas. "Nvidia Geforce RTX 2060 Super review." https://www.techradar.com/nz/reviews/nvidia-geforce-rtx-2060-super (accessed 2 December, 2019).
[175] "List of Nvidia graphics processing units." https://en.wikipedia.org/wiki/List_of_Nvidia_graphics_processing_units#GeForce_256_series (accessed 2 December, 2019).
[176] M. Abadi *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265-283.
[177] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014: ACM, pp. 675-678.
[178] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, 2019, pp. 8024-8035.
[179] A. Gulli and S. Pal, *Deep Learning with Keras*. Packt Publishing Ltd, 2017.
[180] F. Seide and A. Agarwal, "CNTK: Microsoft's open-source deep-learning toolkit," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016: ACM, pp. 2135-2135.
[181] T. T. D. Team *et al.*, "Theano: A Python framework for fast computation of mathematical expressions," *arXiv preprint arXiv:1605.02688,* 2016.
[182] N. Ploskas and N. Samaras, *GPU Programming in Matlab*. Morgan Kaufmann, 2016.
[183] J. Reese and S. Zaranek, "Gpu programming in matlab," *MathWorks News&Notes. Natick, MA: The MathWorks Inc,* pp. 22-5, 2012.
[184] TheEconomist, "Special Report: Artificial Intelligence." [Online]. Available: http://www.economist.com/sites/default/files/ai_mailout.pdf
[185] CB Insights. "AI In Numbers: Global Funding, Exits, And R&D Trends In Artificial Intelligence." https://www.cbinsights.com/research/report/ai-in-numbers-q2-2019/ (accessed 20 November, 2019).
[186] A. Patil, "Artificial Intelligence (AI) Market by Technology (Machine Learning, Natural Language Processing, Image Processing, Speech Recognition), and Industry Vertical (Media & Advertising, BFSI, IT & Telecom, Retail, Healthcare, Automotive & Transportation, and Others) - Global Opportunity Analysis and Industry Forecast, 2018-2025," Allied Market Research, 2018.
[187] Fortune Business Insights, "Artificial Intelligence (AI) Market by Technology (Machine Learning, Natural Language Processing, Image Processing, Speech Recognition), and Industry Vertical (Media & Advertising, BFSI, IT & Telecom, Retail, Healthcare, Automotive & Transportation, and Others) - Global Opportunity Analysis and Industry Forecast, 2018-2025," 2019.
[188] Technology Investment Network, "The Investor's Guide to the New Zealand Technology Sector," Ministry of Business, Innovation & Employment,, 2019. [Online]. Available: https://www.mbie.govt.nz/about/news/investors-guide-to-new-zealands-technology-sector/
[189] NZ Tech, "NZTech Annual Report," 2019, vol. 2019. [Online]. Available: https://nztech.org.nz/reports/nztech-annual-report-2019/
[190] AI Forum New Zealand, "Artificial Intelligence: Shaping a Future New Zealand," 2018. [Online]. Available: https://aiforum.org.nz/reports/artificial-intelligence-shaping-a-future-new-zealand/
[191] G. Rebala, A. Ravi, and S. Churiwala, *An Introduction to Machine Learning*. Springer International Publishing, 2019.
[192] Y. Lin *et al.*, "Large-scale image classification: fast feature extraction and svm training," in *CVPR 2011*, 2011: IEEE, pp. 1689-1696.
[193] J. Sánchez and F. Perronnin, "High-dimensional signature compression for large-scale image classification," in *CVPR 2011*, 2011: IEEE, pp. 1665-1672.

[194] N. Gunji *et al.*, "The Univ. of Tokyo, ILSVRC2012 Scalable Multiclass Object Categorization with Fisher Based Features," n.d.
[195] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*, 2014: Springer, pp. 818-833.
[196] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1-9.
[197] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.
[198] O. Russakovsky *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision,* vol. 115, no. 3, pp. 211-252, 2015.
[199] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132-7141.
[200] P. Clark, J. Thompson, H. Holmback, and L. Duncan, "Exploiting a thesaurus-based semantic net for knowledge-based search," in *AAAI/IAAI*, 2000, pp. 988-995.
[201] C. Fellbaum, "English verbs as a semantic net," *International Journal of Lexicography,* vol. 3, no. 4, pp. 278-301, 1990.
[202] L. Weigang, C. J. P. Alves, and N. Omar, "An expert system for air traffic flow management," *Journal of Advanced Transportation,* vol. 31, no. 3, pp. 343-361, 1997.
[203] F. Baader, R. Küsters, A. Borgida, and D. L. McGuinness, "Matching in description logics," *Journal of Logic and Computation,* vol. 9, no. 3, pp. 411-447, 1999.
[204] P. H. Winston, "Artificial Intelligence," ed: MITOPENCOURSEWARE, 2010.
[205] P. H. Winston, *Artificial Intelligence*, 3rd ed. Boston, MA, USA: Addison-Wesley Longmann Publishing Co., 1992.
[206] S. Raschka and V. Mirjalili, *Python Machine Learning*, 2nd ed. Packt Publishing Ltd, 2017.
[207] J. Chapmann, *Machine Learning Algorithms: Fundamental algorithms for supervised and unsupervised learning with real-world applications*. CreateSpace Independent Publishing Platform, 2017.
[208] M. Wu and Z. Zhang, "Handwritten digit classification using the mnist data set," *Course project CSE802: Pattern Classification & Analysis,* 2010.
[209] J. O. Wao, K. Bivins, R. Hunt III, R. Ries, and S. Schattner, "SAT and ACT scores as predictors of undergraduate GPA scores of construction science and management students," in *Proceedings of the Associated Schools of Construction, 53rd Annual International Conference, Seattle, Washington*, 2017, pp. 5-7.
[210] O. François and E. Durand, "Spatially explicit Bayesian clustering models in population genetics," *Molecular Ecology Resources,* vol. 10, no. 5, pp. 773-784, 2010.
[211] M. S. Dasila, "Get the basics of Machine learning...", ed, 2019.
[212] K. Yau, K.-P. Chow, S.-M. Yiu, and C.-F. Chan, "Detecting anomalous behavior of PLC using semi-supervised machine learning," in *2017 IEEE Conference on Communications and Network Security (CNS)*, 2017: IEEE, pp. 580-585.
[213] A. Kanawaday and A. Sane, "Machine learning for predictive maintenance of industrial machines using iot sensor data," in *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, 2017: IEEE, pp. 87-90.
[214] I. Amihai, R. Gitzel, A. M. Kotriwala, D. Pareschi, S. Subbiah, and G. Sosale, "An Industrial Case Study Using Vibration Data and Machine Learning to Predict Asset Health," in *2018 IEEE 20th Conference on Business Informatics (CBI)*, 2018, vol. 1: IEEE, pp. 178-185.
[215] Google Inc. "AI & Machine Learning Products: AI Platform." https://cloud.google.com/ai-platform/ (accessed 9 December, 2019).

[216] W. Samek, *Explainable AI: Interpreting, explaining and visualizing deep learning*. Springer Nature, 2019.
[217] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, 2017, pp. 4765-4774.
[218] W. Samek, T. Wiegand, and K.-R. Müller, "Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models," *arXiv preprint arXiv:1708.08296,* 2017.
[219] G. Montavon, W. Samek, and K.-R. Müller, "Methods for interpreting and understanding deep neural networks," *Digital Signal Processing,* vol. 73, pp. 1-15, 2018.
[220] R. Szeliski, *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.
[221] H. Barrow, J. Tenenbaum, A. Hanson, and E. Riseman, *Computer vision systems*. 1978.
[222] P. H. Winston and B. Horn, *The psychology of computer vision*. McGraw-Hill Companies, 1975.
[223] R. A. Kumar, V. S. Rajpurohit, and V. B. Nargund, "A neural network assisted machine vision system for sorting pomegranate fruits," in *2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, 22-24 Feb. 2017 2017, pp. 1-9, doi: 10.1109/ICECCT.2017.8118046.
[224] D. Martin, "A practical guide to machine vision lighting," *Retrieved,* vol. 11, no. 05, p. 2013, 2012.
[225] Cognex, "Introduction to Machine Vision: A guide to automating process & quality improvements." [Online]. Available: https://www.assemblymag.com/ext/resources/White_Papers/Sep16/Introduction-to-Machine-Vision.pdf
[226] K. Xia and Z. Weng, "Workpieces sorting system based on industrial robot of machine vision," in *2016 3rd International Conference on Systems and Informatics (ICSAI)*, 2016: IEEE, pp. 422-426.
[227] L. Peilin, Y. Zhen, Z. Wenlong, and L. Hong, "An automatic sorting system for sorting metal cylindrical workpiece based on machine vision and PLC technology," in *2017 2nd International Conference on Robotics and Automation Engineering (ICRAE)*, 2017: IEEE, pp. 446-450.
[228] C. Su, H. Pang, Y. Liu, and J. Ye, "Design of Multi-CPU Automatic Sorting System Based on Machine Vision," in *2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, 2019: IEEE, pp. 209-212.
[229] S. N. Kulkarni and S. K. Singh, "Object Sorting Automated System using Raspberry Pi," in *2018 3rd International Conference on Communication and Electronics Systems (ICCES)*, 2018: IEEE, pp. 217-220.
[230] F. Pedreschi, J. Leon, D. Mery, and P. Moyano, "Development of a computer vision system to measure the color of potato chips," *Food Research International,* vol. 39, no. 10, pp. 1092-1098, 2006.
[231] M.-h. Zhao, "Intelligent Sorting System of Coal Gangue with Machine Vision," in *2018 10th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, 2018, vol. 2: IEEE, pp. 4-7.
[232] J. V. V. Jacques J.P., "The measure of solubility of instant whole milk powder through the use of image processing techniques," Massey University, School of Engineering and Advanced Technology, Palmerston North, New Zealand, 2015.
[233] R. Laszlo, R. Holonec, R. Copîndean, and F. Dragan, "Sorting System for e-Waste Recycling using Contour Vision Sensors," in *2019 8th International Conference on Modern Power Systems (MPS)*, 2019: IEEE, pp. 1-4.

[234] M. Stein, S. Bargoti, and J. Underwood, "Image based mango fruit detection, localisation and yield estimation using multiple view geometry," *Sensors,* vol. 16, no. 11, p. 1915, 2016.
[235] H. S. BAWEJA, T. Parhar, and S. Nuske, "Early-season vineyard shoot and leaf estimation using computer vision techniques," in *2017 ASABE Annual International Meeting*, 2017: American Society of Agricultural and Biological Engineers, p. 1.
[236] E. Anjna and E. R. Kaur, "Review of Image Segmentation Technique," *International Journal,* vol. 8, no. 4, 2017.
[237] S. Yuheng and Y. Hao, "Image Segmentation Algorithms Overview," *arXiv preprint arXiv:1707.02051,* 2017.
[238] I. Aydin, M. Karakose, G. G. Hamsin, A. Sarimaden, and E. Akin, "A new object detection and classification method for quality control based on segmentation and geometric features," in *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*, 16-17 Sept. 2017 2017, pp. 1-6, doi: 10.1109/IDAP.2017.8090172.
[239] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics,* vol. 9, no. 1, pp. 62-66, 1979.
[240] M. Sezgin and B. Sankur, "Survey over image thresholding techniques and quantitative performance evaluation," *Journal of Electronic imaging,* vol. 13, no. 1, pp. 146-166, 2004.
[241] D. G. Lowe, "Object recognition from local scale-invariant features," in *iccv*, 1999, vol. 99, no. 2, pp. 1150-1157.
[242] C. G. Harris and M. Stephens, "A combined corner and edge detector," in *Alvey vision conference*, 1988, vol. 15, no. 50: Citeseer, pp. 10-5244.
[243] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European conference on computer vision*, 2006: Springer, pp. 430-443.
[244] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*, 2006: Springer, pp. 404-417.
[245] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence,* no. 6, pp. 679-698, 1986.
[246] I. Sobel and G. Feldman, "A 3x3 isotropic gradient operator for image processing," *a talk at the Stanford Artificial Project in,* pp. 271-272, 1968.
[247] R. E. Woods and R. C. Gonzalez, *Digital Image Processing*, 4 ed. New York: NY: Pearson, 2018.
[248] A. Allan. "Introducing the NVIDIA Jetson Nano." Hackster.io. https://www.hackster.io/news/introducing-the-nvidia-jetson-nano-aaa9738ef3ff (accessed 17 December, 2019).
[249] Google Inc. "Dev Board." https://coral.ai/products/dev-board/ (accessed 17 November, 2019).
[250] S. Puttemans and T. Goedemé, "Visual detection and species classification of orchid flowers," in *2015 14th IAPR International Conference on Machine Vision Applications (MVA)*, 18-22 May 2015 2015, pp. 505-509, doi: 10.1109/MVA.2015.7153241.
[251] S. Lee, "Deep learning of submerged body images from 2D sonar sensor based on convolutional neural network," in *2017 IEEE Underwater Technology (UT)*, 2017: IEEE, pp. 1-3.
[252] A. Ashiquzzaman and A. K. Tushar, "Handwritten Arabic numeral recognition using deep learning neural networks," in *Imaging, Vision & Pattern Recognition (icIVPR), 2017 IEEE International Conference on*, 2017: IEEE, pp. 1-4.
[253] K. Hsiao, P. Nangeroni, M. Huber, A. Saxena, and A. Y. Ng, "Reactive grasping using optical proximity sensors," in *2009 IEEE International Conference on Robotics and Automation*, 2009: IEEE, pp. 2098-2105.
[254] C. Jing, J. Potgieter, F. Noble, and R. Wang, "A comparison and analysis of RGB-D cameras' depth performance for robotics application," in *2017 24th International*

Conference on Mechatronics and Machine Vision in Practice (M2VIP), 21-23 Nov. 2017 2017, pp. 1-6, doi: 10.1109/M2VIP.2017.8211432.
[255] F. Pece, J. Kautz, and T. Weyrich, "Three depth-camera technologies compared," in *First BEAMING Workshop, Barcelona*, 2011, vol. 2011, p. 9.
[256] F. Alhwarin, A. Ferrein, and I. Scholl, "IR stereo kinect: improving depth images by combining structured light with IR stereo," in *Pacific Rim International Conference on Artificial Intelligence*, 2014: Springer, pp. 409-421.
[257] M. Zollhöfer, "Commodity RGB-D Sensors: Data Acquisition," *arXiv preprint arXiv:1902.06835,* 2019.
[258] P. B. Scott, "The 'Omnigripper': A form of robot universal gripper," *Robotica,* vol. 3, no. 3, pp. 153-158, 1985.
[259] H. Van der Loos, "Design of a three fingered robot gripper," *Industrial Robot: An International Journal,* vol. 5, no. 4, pp. 179-182, 1978.
[260] H. Vanbrussel, B. Santoso, and D. Reynaerts, "Design and control of a multi-fingered robot hand provided with tactile feedback," 1989.
[261] A. Caffaz and G. Cannata, "The design and development of the DIST-Hand dextrous gripper," in *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146)*, 1998, vol. 3: IEEE, pp. 2075-2080.
[262] A. Caffaz, G. Casalino, G. Cannata, G. Panin, and E. Massucco, "The DIST-hand, an anthropomorphic, fully sensorized dexterous gripper," *IEEE Humanoids,* 2000.
[263] J. M. Hollerbach and S. C. Jacobsen, "Anthropomorphic robots and human interactions," in *Proc. of the 1st International Symposium on Humanoid Robots*, 1996: Citeseer, pp. 83-91.
[264] G. C. Causey and R. D. Quinn, "Gripper design guidelines for modular manufacturing," in *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146)*, 1998, vol. 2: IEEE, pp. 1453-1458.
[265] K.-C. Chan and N. C. Cheung, "Grasping of delicate objects by a novel two-finger variable reluctance gripper," in *Conference Record of the 2001 IEEE Industry Applications Conference. 36th IAS Annual Meeting (Cat. No. 01CH37248)*, 2001, vol. 3: IEEE, pp. 1969-1974.
[266] J. Spiliotopoulos, G. Michalos, and S. Makris, "A reconfigurable gripper for dexterous manipulation in flexible assembly," *Inventions,* vol. 3, no. 1, p. 4, 2018.
[267] C.-H. Xiong, W.-R. Chen, B.-Y. Sun, M.-J. Liu, S.-G. Yue, and W.-B. Chen, "Design and implementation of an anthropomorphic hand for replicating human grasping functions," *IEEE Transactions on Robotics,* vol. 32, no. 3, pp. 652-671, 2016.
[268] L. Biagiotti, F. Lotti, C. Melchiorri, and G. Vassura, "How far is the human hand? a review on anthropomorphic robotic end-effectors," 2004.
[269] G. Ponraj Joseph Vedhagiri, A. V. Prituja, C. Li, G. Zhu, N. V. Thakor, and H. Ren, "Pinch grasp and suction for delicate object manipulations using modular anthropomorphic robotic gripper with soft layer enhancements," *Robotics,* vol. 8, no. 3, p. 67, 2019.
[270] Z. Zhang, T. Han, J. Pan, and Z. Wang, "CATCH-919 Hand: Design of a 9-actuator 19-DOF Anthropomorphic Robotic Hand," *arXiv preprint arXiv:1809.04290,* 2018.
[271] I. Llop-Harillo, A. Pérez-González, J. Starke, and T. Asfour, "The Anthropomorphic Hand Assessment Protocol (AHAP)," *Robotics and Autonomous Systems,* vol. 121, p. 103259, 2019.
[272] I. Staretu and C. Moldovan, "Leap motion device used to control a real anthropomorphic gripper," *International Journal of Advanced Robotic Systems,* vol. 13, no. 3, p. 113, 2016.
[273] C. C. Moldovan and I. Staretu, "Capturing human hand movements with a webcam to control an anthropomorphic gripper," *Procedia Manufacturing,* vol. 22, pp. 519-526, 2018.
[274] I. Gaiser *et al.*, "A new anthropomorphic robotic hand," in *Humanoids 2008-8th IEEE-RAS International Conference on Humanoid Robots*, 2008: IEEE, pp. 418-422.

[275] M. Polishchuk, "Anthropomorphic gripping device for an industrial robot: design and calculation of parameters," *SN Applied Sciences,* vol. 1, no. 5, p. 503, 2019.
[276] M. Mudigonda, P. Agrawal, M. Deweese, and J. Malik, "Investigating deep reinforcement learning for grasping objects with an anthropomorphic hand," 2018.
[277] P. C. Huang, J. Lehman, A. K. Mok, R. Miikkulainen, and L. Sentis, "Grasping novel objects with a dexterous robotic hand through neuroevolution," in *2014 IEEE Symposium on Computational Intelligence in Control and Automation (CICA)*, 9-12 Dec. 2014 2014, pp. 1-8, doi: 10.1109/CICA.2014.7013242.
[278] D. Fischinger, A. Weiss, and M. Vincze, "Learning grasps with topographic features," *The International Journal of Robotics Research,* vol. 34, no. 9, pp. 1167-1194, 2015.
[279] V. Babin and C. Gosselin, "Picking, grasping, or scooping small objects lying on flat surfaces: A design approach," *The International Journal of Robotics Research,* vol. 37, no. 12, pp. 1484-1499, 2018.
[280] D. Liang, W. Zhang, Z. Sun, and Q. Chen, "PASA finger: a novel parallel and self-adaptive underactuated finger with pinching and enveloping grasp," in *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2015: IEEE, pp. 1323-1328.
[281] S. Kamada, G. Obinata, and D. Stefanov, "A study of the underactuated mechanisms with compliance," *Adv. Robot. Autom,* vol. 2, no. 1, p. 1, 2013.
[282] H. Fu and W. Zhang, "The Development of a Soft Robot Hand with Pin-Array Structure," *Applied Sciences,* vol. 9, no. 5, p. 1011, 2019.
[283] A. Mo and W. Zhang, "A novel universal gripper based on meshed pin array," *International Journal of Advanced Robotic Systems,* vol. 16, no. 2, p. 1729881419834781, 2019.
[284] H. Fu, H. Yang, W. Song, and W. Zhang, "A novel cluster-tube self-adaptive robot hand," *Robotics and biomimetics,* vol. 4, no. 1, p. 25, 2017.
[285] B. Liang and W. Zhang, "CSB-II Hand: A Universal Griper with Circular Sliding Bars," in *2019 IEEE 4th International Conference on Advanced Robotics and Mechatronics (ICARM)*, 2019: IEEE, pp. 727-731.
[286] J. R. Amend, E. Brown, N. Rodenberg, H. M. Jaeger, and H. Lipson, "A positive pressure universal gripper based on the jamming of granular material," *IEEE Transactions on Robotics,* vol. 28, no. 2, pp. 341-350, 2012.
[287] S. Licht, E. Collins, D. Ballat-Durand, and M. Lopes-Mendes, "Universal jamming grippers for deep-sea manipulation," in *OCEANS 2016 MTS/IEEE Monterey*, 2016: IEEE, pp. 1-5.
[288] Y. Jiang, J. R. Amend, H. Lipson, and A. Saxena, "Learning hardware agnostic grasps for a universal jamming gripper," in *2012 IEEE International Conference on Robotics and Automation*, 2012: IEEE, pp. 2385-2391.
[289] Y. Okatani, T. Nishida, and K. Tadakuma, "Development of universal robot gripper using MRα fluid," in *2014 Joint 7th International Conference on Soft Computing and Intelligent Systems (SCIS) and 15th International Symposium on Advanced Intelligent Systems (ISIS)*, 2014: IEEE, pp. 231-235.
[290] J. Kapadia and M. Yim, "Design and performance of nubbed fluidizing jamming grippers," in *2012 IEEE International Conference on Robotics and Automation*, 2012: IEEE, pp. 5301-5306.
[291] N. G. Cheng *et al.*, "Design and analysis of a robust, low-cost, highly articulated manipulator enabled by jamming of granular media," in *2012 IEEE International Conference on Robotics and Automation*, 2012: IEEE, pp. 4328-4333.
[292] K. Harada *et al.*, "Proposal of a shape adaptive gripper for robotic assembly tasks," *Advanced Robotics,* vol. 30, no. 17-18, pp. 1186-1198, 2016.
[293] Y. Hao *et al.*, "Universal soft pneumatic robotic gripper with variable effective length," in *2016 35th Chinese Control Conference (CCC)*, 2016: IEEE, pp. 6109-6114.

[294] M. E. Salem, Q. Wang, R. Wen, and M. Xiang, "Design and Characterization of Soft Pneumatic Actuator for Universal Robot Gripper," in *2018 International Conference on Control and Robots (ICCR)*, 2018: IEEE, pp. 6-10.
[295] J. Ponce and B. Faverjon, "On computing three-finger force-closure grasps of polygonal objects," *IEEE Transactions on Robotics and Automation,* vol. 11, no. 6, pp. 868-881, 1995, doi: 10.1109/70.478433.
[296] A. Morales, E. Chinellato, P. Sanz, A. Del Pobil, and A. H. Fagg, "Learning to predict grasp reliability for a multifinger robot hand by using visual features," *AISC proceedings,* 2004.
[297] E. Chinellato, R. B. Fisher, A. Morales, and A. P. Del Pobil, "Ranking planar grasp configurations for a three-finger hand," in *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, 2003, vol. 1: IEEE, pp. 1133-1138.
[298] A. Bicchi and V. Kumar, "Robotic grasping and contact: a review," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, 24-28 April 2000 2000, vol. 1, pp. 348-353 vol.1, doi: 10.1109/ROBOT.2000.844081.
[299] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L. J. Guibas, "Normalized Object Coordinate Space for Category-Level 6D Object Pose and Size Estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2642-2651.
[300] J. M. Wong *et al.*, "Segicp: Integrated deep semantic segmentation and pose estimation," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017: IEEE, pp. 5784-5789.
[301] A. Collet, M. Martinez, and S. Srinivasa, *The MOPED framework: Object recognition and pose estimation for manipulation*. 2011, pp. 1284-1306.
[302] M. V. Liarokapis and A. M. Dollar, "Learning task-specific models for dexterous, in-hand manipulation with simple, adaptive robot hands," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016: IEEE, pp. 2534-2541.
[303] R. Strudel, A. Pashevich, I. Kalevatykh, I. Laptev, J. Sivic, and C. Schmid, "Combining learned skills and reinforcement learning for robotic manipulations," *arXiv preprint arXiv:1908.00722,* 2019.
[304] K. Fang *et al.*, "Learning task-oriented grasping for tool manipulation from simulated self-supervision," *The International Journal of Robotics Research,* p. 0278364919872545, 2019.
[305] Z. Qin, K. Fang, Y. Zhu, L. Fei-Fei, and S. Savarese, "KETO: Learning Keypoint Representations for Tool Manipulation," *arXiv preprint arXiv:1910.11977,* 2019.
[306] PassMark. "NVIDIA TITAN Xp." https://www.videocardbenchmark.net/gpu.php?gpu=NVIDIA+TITAN+Xp&id=3728 (accessed 21 February, 2020).
[307] Kawasaki Heavy Industries Ltd. *Kawasaki E Series EX100*. (1982). [Online]. Available: http://kri-us.com/PDFs/EX100.pdf
[308] Kawasaki Heavy Industries Ltd. *Standard Specifications: BX100SFE02*. (2015). [Online]. Available: https://robotics.kawasaki.com/userAssets1/productPDF/BX100SFE02-E.pdf
[309] M. T. Mason, "Toward robotic manipulation," *Annual Review of Control, Robotics, and Autonomous Systems,* vol. 1, pp. 1-28, 2018.
[310] A. Billard and D. Kragic, "Trends and challenges in robot manipulation," *Science,* vol. 364, no. 6446, p. eaat8414, 2019.
[311] K. B. Shimoga, "Robot grasp synthesis algorithms: A survey," *The International Journal of Robotics Research,* vol. 15, no. 3, pp. 230-266, 1996.
[312] R. V. Bostelman and J. A. Falco, "Survey of industrial manipulation technologies for autonomous assembly applications," 2012.

[313] J. Sanchez, J.-A. Corrales, B.-C. Bouzgarrou, and Y. Mezouar, "Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey," *The International Journal of Robotics Research,* vol. 37, no. 7, pp. 688-716, 2018.
[314] A. M. Okamura, N. Smaby, and M. R. Cutkosky, "An overview of dexterous manipulation," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, 2000, vol. 1: IEEE, pp. 255-262.
[315] D. Prattichizzo and J. Trinkle, "Handbook of robotics, chapter Grasping," ed: Springer, Heidelberg, 2008.
[316] R. M. Murray, *A mathematical introduction to robotic manipulation*. CRC press, 1994.
[317] V.-D. Nguyen, "Constructing force-closure grasps," *The International Journal of Robotics Research,* vol. 7, no. 3, pp. 3-16, 1988.
[318] C. Ferrari and J. F. Canny, "Planning optimal grasps," in *ICRA*, 1992, vol. 3, pp. 2290-2295.
[319] R. D. Hester, M. Cetin, C. Kapoor, and D. Tesar, "A criteria-based approach to grasp synthesis," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, 10-15 May 1999 1999, vol. 2, pp. 1255-1260 vol.2, doi: 10.1109/ROBOT.1999.772533.
[320] Y. C. Park and G. P. Starr, "Grasp synthesis of polygonal objects," in *Proceedings., IEEE International Conference on Robotics and Automation*, 13-18 May 1990 1990, pp. 1574-1580 vol.3, doi: 10.1109/ROBOT.1990.126233.
[321] D. L. Bowers and R. Lumia, "Manipulation of unmodeled objects using intelligent grasping schemes," *IEEE Transactions on Fuzzy Systems,* vol. 11, no. 3, pp. 320-330, 2003.
[322] R. Paolini, A. Rodriguez, S. S. Srinivasa, and M. T. Mason, "A data-driven statistical framework for post-grasp manipulation," *The International Journal of Robotics Research,* vol. 33, no. 4, pp. 600-615, 2014.
[323] C. Neef, D. Luipers, J. Bollenbacher, C. Gebel, and A. Richert, "Towards Intelligent Pick and Place Assembly of Individualized Products Using Reinforcement Learning," EasyChair, 2516-2314, 2020.
[324] P. Jiang *et al.*, "Depth Image–Based Deep Learning of Grasp Planning for Textureless Planar-Faced Objects in Vision-Guided Robotic Bin-Picking," *Sensors,* vol. 20, no. 3, p. 706, 2020.
[325] P. Shukla, H. Kumar, and G. Nandi, "Robotic Grasp Manipulation Using Evolutionary Computing and Deep Reinforcement Learning," *arXiv preprint arXiv:2001.05443,* 2020.
[326] H. Liu and C. Cao, "Grasp Pose Detection Based On Point Cloud Shape Simplification," in *IOP Conference Series: Materials Science and Engineering*, 2020, vol. 717, no. 1: IOP Publishing, p. 012007.
[327] J. Mahler *et al.*, "Learning ambidextrous robot grasping policies," *Science Robotics,* vol. 4, no. 26, p. eaau4984, 2019.
[328] X. Xie, C. Li, C. Zhang, Y. Zhu, and S.-C. Zhu, "Learning virtual grasp with failed demonstrations via bayesian inverse reinforcement learning," 2019: IROS.
[329] Y. Li, L. Schomaker, and S. H. Kasaei, "Learning to Grasp 3D Objects using Deep Residual U-Nets," *arXiv preprint arXiv:2002.03892,* 2020.
[330] S. V. Pharswan, M. Vohra, A. Kumar, and L. Behera, "Domain Independent Unsupervised Learning to grasp the Novel Objects," *arXiv preprint arXiv:2001.05856,* 2020.
[331] M. Mahajan, T. Bhattacharjee, A. Krishnan, P. Shukla, and G. Nandi, "Semi-supervised Grasp Detection by Representation Learning in a Vector Quantized Latent Space," *arXiv preprint arXiv:2001.08477,* 2020.

[332] O. Kroemer, S. Niekum, and G. Konidaris, "A review of robot learning for manipulation: Challenges, representations, and algorithms," *arXiv preprint arXiv:1907.03146,* 2019.
[333] D. Guo, F. Sun, H. Liu, T. Kong, B. Fang, and N. Xi, "A hybrid deep architecture for robotic grasp detection," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 29 May-3 June 2017 2017, pp. 1609-1614, doi: 10.1109/ICRA.2017.7989191.
[334] F. Sadeghi, A. Toshev, E. Jang, and S. Levine, "Sim2real viewpoint invariant visual servoing by recurrent control," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4691-4699.
[335] C. P. Bechlioulis, S. Heshmati-alamdari, G. C. Karras, and K. J. Kyriakopoulos, "Robust image-based visual servoing with prescribed performance under field of view constraints," *IEEE Transactions on Robotics,* vol. 35, no. 4, pp. 1063-1070, 2019.
[336] A. X. Lee, S. Levine, and P. Abbeel, "Learning visual servoing with deep features and fitted q-iteration," *arXiv preprint arXiv:1703.11000,* 2017.
[337] Z. Zake, S. Caro, A. S. Roos, F. Chaumette, and N. Pedemonte, "Stability Analysis of Pose-Based Visual Servoing Control of Cable-Driven Parallel Robots," in *International Conference on Cable-Driven Parallel Robots*, 2019: Springer, pp. 73-84.
[338] K. M. Nalini and R. R. Gondkar, "Roboic recognition for unstructered 2-D parts to pick and place objects," in *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, 19-20 May 2017 2017, pp. 1478-1482, doi: 10.1109/RTEICT.2017.8256843.
[339] Insight. "NVIDIA Tesla K20 - GPU computing processor - Tesla K20." https://www.insight.com/en_US/shop/product/UCSC-GPU-K20=/Cisco/UCSC-GPU-K20=/NVIDIATeslaK20-GPUcomputi/ (accessed 21 February, 2020).
[340] Versus.com. "Nvidia GeForce GTX 645 OEM review: specs and price." https://versus.com/en/nvidia-geforce-gtx-645-oem (accessed 21 February, 2020).
[341] P. Triantafyllou, H. Mnyusiwalla, P. Sotiropoulos, M. A. Roa, D. Russell, and G. Deacon, "A benchmarking framework for systematic evaluation of robotic pick-and-place systems in an industrial grocery setting," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019: IEEE, pp. 6692-6698.
[342] S. Cruciani, B. Sundaralingam, K. Hang, V. Kumar, T. Hermans, and D. Kragic, "Benchmarking In-Hand Manipulation," *arXiv preprint arXiv:2001.03070,* 2020.
[343] B. Yang, P. Lancaster, S. Srinivasa, and J. R. Smith, "Benchmarking Robot Manipulation with the Rubik's Cube," *IEEE Robotics and Automation Letters,* 2020.
[344] A. H. Quispe, H. B. Amor, and H. I. Christensen, "A taxonomy of benchmark tasks for robot manipulation," in *Robotics Research*: Springer, 2018, pp. 405-421.
[345] European Robotics Research Network. "Euron GEM Sig." http://www.heronrobots.com/EuronGEMSig/ (accessed 25 February, 2020).
[346] IEEE Robotics & Automation Society. "Technical committee for performance evaluation & benchmarking of robotic and automation systems." https://www.ieee-ras.org/performance-evaluation (accessed 25 February, 2020).
[347] Defence Advanced Research Projects Agency. "Autonomous Robotic Manipulation (ARM)." https://www.darpa.mil/program/autonomous-robotic-manipulation (accessed 25 February, 2020).
[348] Intelligent Robots and Systems. "Call for Competitors." https://www.iros2019.org/call-for-competitions (accessed 25 February, 2020).
[349] B. Yang, J. Zhang, V. Pong, S. Levine, and D. Jayaraman, "Replab: A reproducible low-cost arm benchmark platform for robotic learning," *arXiv preprint arXiv:1905.07447,* 2019.

[350] F. Bottarel, G. Vezzani, U. Pattacini, and L. Natale, "GRASPA 1.0: GRASPA is a robot arm grasping performance benchmark," *IEEE Robotics and Automation Letters,* vol. 5, no. 2, pp. 836-843, 2020.
[351] A. Murali *et al.*, "Pyrobot: An open-source robotics framework for research and benchmarking," *arXiv preprint arXiv:1906.08236,* 2019.
[352] L. Jamone, A. Bernardino, and J. Santos-Victor, "Benchmarking the grasping capabilities of the iCub hand with the YCB Object and Model Set," *IEEE Robotics and Automation Letters,* vol. 1, no. 1, pp. 288-294, 2016.
[353] Y. S. Choi, T. Deyle, and C. C. Kemp, "A list of household objects for robotic retrieval prioritized by people with ALS (Version 092008)," *arXiv preprint arXiv:0902.2186,* 2009.
[354] A. Tow. "APB Shopping List." https://docs.google.com/spreadsheets/d/1HGlDC7FVBJlMGyTyHMESfwngLTUZ8wI6EoCqpCVeixo/edit#gid=1293082008 (accessed 27 February, 2020).
[355] F. T. Pokorny, Y. Bekiroglu, K. Pauwels, J. Bütepage, C. Scherer, and D. Kragic, "CapriDB-Capture, Print, Innovate: A Low-Cost Pipeline and Database for Reproducible Manipulation Research," *arXiv preprint arXiv:1610.05175,* 2016.
[356] C. Goldfeder, M. Ciocarlie, H. Dang, and P. K. Allen, "The columbia grasp database," in *2009 IEEE international conference on robotics and automation*, 2009: IEEE, pp. 1710-1716.
[357] A. Singh, J. Sha, K. Narayan, T. Achim, and P. Abbeel, *BigBIRD: A large-scale 3D database of object instances*. 2014, pp. 509-516.
[358] G. Georgakis, M. A. Reza, A. Mousavian, P.-H. Le, and J. Košecká, "Multiview RGB-D dataset for object instance detection," in *2016 Fourth International Conference on 3D Vision (3DV)*, 2016: IEEE, pp. 426-434.
[359] A. Kasper, Z. Xue, and R. Dillmann, "The KIT object models database: An object model database for object recognition, localization and manipulation in service robotics," *The International Journal of Robotics Research,* vol. 31, no. 8, pp. 927-934, 2012.
[360] D. Pickup *et al.*, "SHREC'14 track: Shape retrieval of non-rigid 3D human models," in *Proceedings of the 7th Eurographics workshop on 3D Object Retrieval*, 2014, vol. 1, no. 2: Eurographics Association, p. 6.
[361] C. Rennie, R. Shome, K. E. Bekris, and A. F. De Souza, "A dataset for improved rgbd-based object detection and pose estimation for warehouse pick-and-place," *IEEE Robotics and Automation Letters,* vol. 1, no. 2, pp. 1179-1185, 2016.
[362] M. Ciocarlie, C. Pantofaru, K. Hsiao, G. Bradski, P. Brook, and E. Dreyfuss, "A side of data with my robot," *IEEE Robotics & Automation Magazine,* vol. 18, no. 2, pp. 44-57, 2011.
[363] A. Miller, P. Allen, V. Santos, and F. Valero-Cuevas, "From robotic hands to human hands: a visualization and simulation engine for grasping research," *Industrial Robot: An International Journal,* 2005.
[364] A. Miller and P. Allen. "GraspIt!" https://graspit-simulator.github.io/ (accessed 2 March, 2020).
[365] B. León *et al.*, "Opengrasp: a toolkit for robot grasping simulation," in *International Conference on Simulation, Modeling, and Programming for Autonomous Robots*, 2010: Springer, pp. 109-120.
[366] R. Diankov and J. Kuffner, "Openrave: A planning architecture for autonomous robotics," *Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-08-34,* vol. 79, 2008.
[367] J.-B. Mouret and K. Chatzilygeroudis, "20 years of reality gap: a few thoughts about simulators in evolutionary robotics," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2017, pp. 1121-1124.

[368] S. Koos, J.-B. Mouret, and S. Doncieux, "Crossing the reality gap in evolutionary robotics by promoting transferable controllers," in *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, 2010, pp. 119-126.
[369] P. Martinez-Gonzalez, S. Oprea, A. Garcia-Garcia, A. Jover-Alvarez, S. Orts-Escolano, and J. Garcia-Rodriguez, "Unrealrox: an extremely photorealistic virtual reality environment for robotics simulations and synthetic data generation," *Virtual Reality,* pp. 1-18, 2019.
[370] J. Collins, D. Howard, and J. Leitner, "Quantifying the reality gap in robotic manipulation tasks," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019: IEEE, pp. 6706-6712.
[371] The Python Software Foundation, "Python: a programming language changes the world," ed, 2015.
[372] Erik Debill. "Module Counts." http://www.modulecounts.com/ (accessed 4 March, 2020).
[373] M. Abadi *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467,* 2016.
[374] S. Van der Walt *et al.*, "scikit-image: image processing in Python," *PeerJ,* vol. 2, p. e453, 2014.
[375] P. Jaccard, *Distribution de la Flore Alpine dans le Bassin des Dranses et dans quelques régions voisines*. 1901, pp. 241-72.
[376] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in *Advances in neural information processing systems*, 2013, pp. 2553-2561.
[377] A. Sobti, C. Arora, and M. Balakrishnan, "Object detection in real-time systems: Going beyond precision," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018: IEEE, pp. 1020-1028.
[378] A. Rosebrock. "What is the Jaccard Index?" https://deepai.org/machine-learning-glossary-and-terms/jaccard-index (accessed 9 March, 2020).
[379] W. Wang, J. Dong, and B. Tieniu Tan, "Position determines perspective: Investigating perspective distortion for image forensics of faces," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 1-9.
[380] S. Patel and J. Topiwala, *Measuring a Centre of Gravity of an Object using 4 Load Transducer Method*. 2017.
[381] D. Tynan, "RTP payload format for BT. 656 Video Encoding," *Request for Comments (Proposed Standard) RFC,* vol. 2431, 1998.
[382] E. S. Gedraite and M. Hadad, "Investigation on the effect of a Gaussian Blur in image filtering and segmentation," in *Proceedings ELMAR-2011*, 2011: IEEE, pp. 393-396.
[383] A. Rahman, F. Yasmin, A. Hussain, W. M. D. Wan Zaki, H. Badioze Zaman, and N. Md Tahir, "Enhancement of background subtraction techniques using a second derivative in gradient direction filter," *Journal of Electrical and Computer Engineering,* vol. 2013, 2013.
[384] D. Suresh and M. Lavanya, "Motion Detection and Tracking using Background Subtraction and Consecutive Frames Difference Method," *International Journal of Research Studies in Science, Engineering and Technology,* vol. 1, no. 5, pp. 16-22, 2014.
[385] E. Hayman and J.-O. Eklundh, "Statistical background subtraction for a mobile observer," in *null*, 2003: IEEE, p. 67.
[386] M. Buckland and F. Gey, "The relationship between recall and precision," *Journal of the American society for information science,* vol. 45, no. 1, pp. 12-19, 1994.
[387] T. Fawcett, "An introduction to ROC analysis," *Pattern recognition letters,* vol. 27, no. 8, pp. 861-874, 2006.

[388] S. Carter, Z. Armstrong, L. Schubert, I. Johnson, and Olah. "Exploring Neural Networks with Activation Atlases." https://distill.pub/2019/activation-atlas/?utm_campaign=Data_Elixir (accessed 28 March, 2020).
[389] S. R. Gunn, "Support vector machines for classification and regression," *ISIS technical report,* vol. 14, no. 1, pp. 5-16, 1998.
[390] A. Reyes, *Beginner's Guide to SolidWorks 2014-Level I*. SDC Publications, 2014.
[391] Dassault Systèmes. "Solidworks Help, Model Display." https://help.solidworks.com/2018/english/SolidWorks/sldworks/c_Model_Display.htm?id=63b1ee36aab84c65b599a3a952ac3610#Pg0 (accessed 2020, 15 April).
[392] NVIDIA Corporation. "NVIDIA QUADRO K2200 Datasheet." https://www.nvidia.com/content/dam/en-zz/Solutions/design-visualization/documents/75509_DS_NV_Quadro_K2200_US_NV_HR.pdf (accessed 16 July, 2019).
[393] Intel Corporation, "Intel® Core™ i7-4790 Processor," 2017. [Online]. Available: https://ark.intel.com/content/www/us/en/ark/products/80806/intel-core-i7-4790-processor-8m-cache-up-to-4-00-ghz.html
[394] The Mathworks Inc. "Products and Services." https://au.mathworks.com/products.html?s_tid=gn_ps (accessed 15 April, 2020).
[395] Shenzhen Yuejiang Technology Co. Ltd, "Dobot Magician User Manual," 2017, issue V1.2.4. [Online]. Available: https://www.generationrobots.com/media/Dobot-Magician-User-Manual-V1.2.4.pdf
[396] Mean Well Enterprises, "60W Single Output Industrial DIN Rail Power Supply MDR-60 series," n.d. [Online]. Available: https://www.meanwell-web.com/content/files/pdfs/productPdfs/MW/Mdr-60/MDR-60-spec.pdf
[397] L. CHINFA ELECTRONICS IND. CO., "AMR1 SERIES," 2014. [Online]. Available: https://docs.rs-online.com/b0cf/0900766b8144d4b2.pdf
[398] Leadshine Technology Co. Ltd., "M542 Economical Microstepping Driver," n.d. [Online]. Available: http://www.leadshine.com/UploadFile/Down/M542d.pdf
[399] K. Hoffmann, *Applying the Wheatstone bridge circuit*. HBM Germany, 1974.
[400] HT Sensor Technology Co. Ltd. "TAL220 Parallel Beam Load Cell." https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/TAL220M4M5Update.pdf (accessed 16 July, 2019).
[401] Avia Semiconductor. "HX711. 24-Bit Analog-to-Digital Converter (ADC) for Weigh Scales." https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/hx711_english.pdf (accessed 16 July, 2019).
[402] CHANGZHOU SONGYANG MACHINERY & ELECTRONICS NEW TECHNIC INSTITUTE, "HIGH TORQUE HYBRID STEPPING MOTOR SPECIFICATIONS," 2012. [Online]. Available: https://www.pololu.com/file/0J629/SY57STH76-2804A.pdf
[403] M. Walter. "HOW ACCURATE IS MICROSTEPPING REALLY?" https://hackaday.com/2016/08/29/how-accurate-is-microstepping-really/ (accessed 16 April, 2020).
[404] G. Beauchemin, "Stepper Motor Technical Note: Microstepping Myths and Realities," MICROMO FAULHABER GROUP, 2003. [Online]. Available: https://www.micromo.com/media/wysiwyg/Technical-library/Stepper/6_Microstepping%20WP.pdf
[405] Atmel, "AVR446: Linear speed control of stepper motor," Rev. 8017A-AVR-06/06, n.d., vol. Rev. 8017A-AVR-06/06.
[406] Microsoft, "Microsoft LifeCam Studio Technical Data Sheet." [Online]. Available: http://download.microsoft.com/download/0/9/5/0952776D-7A26-40E1-80C4-76D73FC729DF/TDS_LifeCamStudio.pdf
[407] RS Components Ltd, "Datasheet: RS Pro 2.5m White LED Strip 136-3579," n.d. [Online]. Available: https://docs.rs-online.com/256e/0900766b815e69e7.pdf

[408] Carlo Gavazzi Holding, "Datasheet: Photoelectrics Diffuse-reflective Type PA18C.D..., DC," 2017. [Online]. Available: https://docs.rs-online.com/906c/0900766b8170a64f.pdf
[409] D. Dechow, "The Fundamentals of Machine Vision." [Online]. Available: https://www.visiononline.org/userAssets/aiaUploads/file/T1-The-Fundamentals-of-Machine-Vision.pdf
[410] Dobot.cc. "DOBOT Magician: Lightweight Intelligent Training Robotic Arm - An all-in-one STEAM Education Platform." https://www.dobot.cc/dobot-magician/product-overview.html (accessed April 17, 2020).
[411] Shenzhen Yuejiang Technology Co Ltd., "Dobot Magician User Guide," 2018, issue V1.5.1. [Online]. Available: https://download.dobot.cc/product-manual/dobot-magician/pdf/V1.5.1/en/Dobot-Magician-User-Guide-V1.5.1.pdf
[412] Shenzhen Yuejiang Technology Co Ltd., "Dobot Magician Demo Description (MATLAB)," 2018, issue V1.0.
[413] Shenzen Yuejiang Technology Co Ltd., "Dobot Magician Communication Protocol," 2018, issue V1.1.3. [Online]. Available: https://jacquesjohnston.files.wordpress.com/2019/01/dobot-magician-communication-protocol-v1.1.3.pdf
[414] J. J. P. Johnston. "Communicating with the Dobot Magician using raw protocol." Wordpress. https://jacquesjohnston.wordpress.com/2019/01/29/communicating-with-the-dobot-magician-using-raw-protocol/ (accessed April 17, 2020).
[415] J. Kilian, "Simple Image Analysis By Moments," 2001, issue 0.2. [Online]. Available: http://breckon.eu/toby/teaching/dip/opencv/SimpleImageAnalysisbyMoments.pdf
[416] N. Corporation. "GEFORCE RTX 2070." https://www.nvidia.com/en-us/geforce/graphics-cards/rtx-2070/ (accessed 7 May, 2020).
[417] Advanced Micro Devices Incorporated. "AMD Ryzen 7 2700 Processor." https://www.amd.com/en/products/cpu/amd-ryzen-7-2700 (accessed 7 May, 2020).

Appendix A Industry perspective

In this thesis, an autonomous object handling process is investigated within the context of flexible and customisable robotic production technologies for manufacture. Integration with existing hardware and practices and general industrial relevance are key considerations of this study.

Although it was recognised that a strong literature-based argument could be made for the utility of this work, an early proposal of this project was criticised for failing to discuss the research with relevant industries. To address this issue, it was suggested to consult with 1-2 groups in the robotic automation and manufacture field. To determine the interest of novel object handling and its pertinence, several process control, automation and engineering-related organisations and businesses based in New Zealand were engaged, including Massey University's School of Food and Advanced Technology, Callaghan Innovation, Control Box, and various other related groups that may not wish to be identified.

To gauge the general sentiment of these industries, broad questions were posed with quantitative aspects in the form of a short survey. Details related to this research and the scope of the project were discussed in person, or via phone, prior to survey involvement. Additional resources were made available, and further discussions conducted, to applicants seeking further information. Some applicants preferred to avoid the survey, opting for a qualitative discussion related to the questions posed in the survey. Due to the anecdotal nature of this investigation and lack of comprehensive response, it is included as an appendix. Three surveys were filled out in full. Approximately 20 formal discussions surrounding the topics covered in the survey were undertaken by phone or in person. The full survey can be found below.

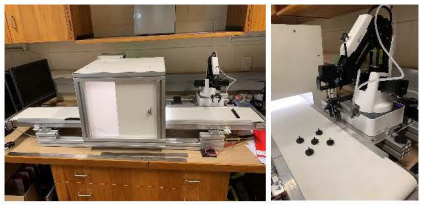
Context

The automated manipulation of objects previously unseen by a robotic system is an extremely difficult task, as a good grasp is related to object shape, size, weight, weight-distribution, centre of mass, surface properties, friction coefficients, object deformability, and can be harshly affected by sensing and actuation precision. Moreover, the relationship between these variables and a specific grasping strategy, robotic hardware and gripper is not always clear. Due to these factors, a robust solution to automatic manipulation is missing. Autonomous novel object grasping research has traditionally been centred around domestic tasks such as clutter clearing, object sorting and dishwasher unloading, although works have started to focus on more industry-related applications. For instance, rubbish sorting, automated checkout robots and warehouse automation—made popular by the Amazon Robotics Challenge.

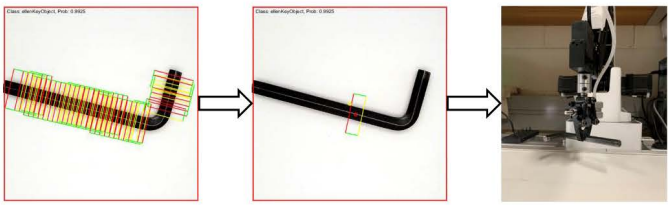
Over the past three decades there has been a shift toward flexible, reconfigurable and automated production systems. Although flexible hardware is progressively becoming a topic of interest—such as the dual-arm concept developed by ABB—the adaptability of the associated control methodology is not usually considered to the same degree. Fully manual assembly lines are still common in low-wage countries, particularly for manufacturers of consumer electronics, small appliances, toys, etc. We believe that a robust, autonomous handling process plays a crucial role in future manufacturing systems and is key for further closing the gap between manual and fully automated production.

Brief project description

At Massey we're working on a scalable, manipulator-agnostic methodology which aims to facilitates both good grasping and good handling of unknown objects. Our methodology employs machine learning coupled with vision and other input modalities to generate, score and select an optimal grasping location for a specific object. Our current implementation makes use of a conveyor-belt and vision enclosure system. Objects are placed haphazardly at one end of the conveyor. Detected objects are automatically moved through the enclosure to the opposite end, where they may be manipulated by a fixed robotic arm without instruction.



Depending on the results, information regarding this object and subsequent grasp is added to a database to help further improve our learning framework. Although our methodology can determine how to best grasp an object, what should be done with that object must still be specified to some degree. Our aim is to enable high-level user instruction, such as place object A at position B. We want to avoid low-level user instruction, such as move arm to location A, open gripper, lower end-effector, close gripper, raise end-effector, move to position B, lower end-effector, open gripper, raise end-effector, etc. Our algorithm attempts to find a stable grasping location. Additionally, how that object is consequently handled and finally placed is also considered. Our research is focused specifically on industrial application and production line integration, intended to be used for handling and assembly of small parts at this stage of development, e.g., consumer electronics.



Disclaimer

The purpose of this survey is to get a feel for the applicability of novel object grasping from people within related areas. Any personal details provided by you will remain confidential. This survey involves recording your responses to questions about our research and research area. Although your specific responses will remain confidential, the overall results of this survey will be associated with the appropriate field. For example, it may be stated that participants within the robotic welding industry felt that automated grasping is not useful within their field. Results of this survey will be used for scholarly purposes. This survey takes approximately 15 minutes. Signing below indicates that: • You have been made aware of the above information • You voluntarily agree to participate • You are at least 18 years of age

Signature	

Basic information

Name	
Job title	
Company	
Years in industry	
Date of interview	

What do you do for your organization?

Research-related

Automated novel object grasping relates to the ability of a robotic manipulator to automatically pick objects observed for the first time, without explicit instruction. In general, how applicable do you think novel object grasping is to manufacture and production? (sorting, dispensing, packing, loading and unloading, pick and place, press fitting, etc.)
(/10)
In general, how applicable is novel object grasping to small part assembly? (small consumer goods, medical equipment, small pneumatic and automotive componentry, small electronics, etc.)
(/10)
Our method currently centers around smaller objects. How scalable do you think our research is? (vision, database, machine learning, conveyor system, etc.)
(/10)

Applicant-related

What types of applications do you usually use robotics for? (research, assembly, packaging, sorting, machining, welding, etc.)
In an average workweek, what percentage of your time is spent with robotics?
(/100)
Of that time, what percentage is spent programming or automating robotic processes?
(/100)
How are these robots programmed? (tick if applicable)
☐ Coding environment / simulation ☐ Pendant ☐ Lead through (teaching by demonstration) Other:

Changeover

Changeover is the process of changing a production line to accommodate for the manufacture of a new product. The time taken to perform changeovers can cause significant losses in production time, especially when robotics is incorporated in the manufacturing process. Robotic changeover refers to the process of preparing a manipulator for a new procedure. Do you have any experience related to robotic changeover?
Do you recognize this as a potential area that can be automated by our area of research?
(/10)
How many people / hours would you say it takes to prepare a robotic system for a new production process?
How useful would you say the ability for a robotic system to automatically adapt to new objects is for reducing changeover time?
(/10)
When setting up a new robotic manufacturing process, how much information regarding that new product is usually available? (please tick if applicable)
☐ 3D model ☐ Image ☐ Physical object Other:

Open-ended

What are your thoughts on the viability of an autonomous object handling system in your field?
What applications do you think would suit automated novel object handling?
Improvements, suggestions or any further comments.

Thank you for participating in this survey!

The first question under the heading *Research-related*, associated with the general applicability of novel object handling to manufacture, received a quantitative response of 8/10. It was noted that applicability is highly dependent on the specific application, system performance, reliability, and degree to which differing objects may be handled. A similar response was given to the following two questions under this heading, resulting in 8/10 for applicability related to grasping small parts and 7/10 for scalability. Some applicants noted that it was difficult to comment on the scalability without in-depth information of the proposed methodology.

Participants were involved in robotics research, prototyping, automation, and general engineering. Applicants recorded that they spend approximately 5-10% of their time working with robotics. Of that time, between 30-90% is spent programming or automating the robotic process. The most common programming methods are sequential coding, lead through and coding environment/simulation.

All applicants were familiar with preparing a robotic process for new procedures under the *Changeover* section. Some applicants did not have experience in a production setting, but were generally involved in new, application-specific setups. Others were exclusively involved in new and existing installations in a manufacturing environment. Given the question related to the changeover automation potential of this work, applicants overwhelmingly recognised that changeover time could be significantly reduced by this research, although, many noted that this would be dependent on the robustness and reliability of the system. The quantitative response to the second question under the *Changeover* heading was 10/10. The quantitative response of the fourth question under the *Changeover* heading was 9/10. Generally, applicants could not provide a specific answer related to the amount of labour required to prepare a robotic system for a new production process. When setting up a new robotic manufacturing process, many applicants noted that usually images, the physical object, 3D models, material properties, cost and many other details regarding the new product are available.

Under the *Open-ended* section, applicants noted that automated novel object handling would be well suited to production lines that handle a variety of objects in different poses. Moreover, such a system would be particularly useful in cases that were subject to frequent change. Generally, the response was that this research could have significant practical utility in a manufacturing environment—but that more information may be needed to better gauge the project in its entirety. Applicants were generally very interested and regarded any form of automation in this context as useful for future industry. Packaging, process control, automation, sorting and production line applications were commonly discussed.

Appendix B ICIEA conference picture

This research produced a conference paper in 2019. Consequently, a presentation was given at the 14th annual IEEE Conference on Industrial Electronics and Applications in Xi'an, China. This conference provided an excellent opportunity to present some of the work related to this research and gauge the response from academics in similar fields. The work was well received, with several beneficial discussions after the presentation. An image related to the conference is shown in Figure B.1.

***Figure B.1*. Image taken in Jianguo Hotel, Xi'an. The author of this thesis, Jacques Janse van Vuuren, is second from the right. The main supervisor of this work, Liqiong Tang, is fifth from the left.**

Appendix C Round 2 trial results

The results of the second round of trials are illustrated in this section. For more information regarding the tested objects, refer to Chapter 5. Figure C.1, C.2, C.3 and C.4 illustrate the resultant OS and OE scores by object series, category and label for random selection, highest $K_{I_{RGW}}$, highest $pass_{pred}$ selection and highest combined OS_{pred}, OE_{pred} selection, respectively. Figure C.5, C.6 and C.7 illustrate the resultant $pass/fail$ label of each object by series, category, and respective selection criterion.

Random selection										
		Household			Tool			Component		
		Pass	Fail	Average	Pass	Fail	Average	Pass	Fail	Average
Known	OS	0.578	0.121	0.550	0.551	0.068	0.522	0.521	0.159	0.492
	OE	0.968	0.483	0.939	0.968	0.521	0.942	0.984	0.719	0.962
Unknown	OS	0.583	0.161	0.564	0.536	0.082	0.499	0.580	0.093	0.548
	OE	0.977	0.714	0.965	0.953	0.751	0.936	0.971	0.476	0.938
Misc.	OS	0.569	0.068	0.527	0.550	0.039	0.535	0.541	0.052	0.520
	OE	0.928	0.535	0.895	0.987	0.443	0.971	0.947	0.506	0.927

***Figure C.1*. Average measured OS and OE scores by series, category, and label for the random selection criterion.**

Highest KI_RGW selection										
		Household			Tool			Component		
		Pass	Fail	Average	Pass	Fail	Average	Pass	Fail	Average
Known	OS	0.644	0.135	0.614	0.532	0.218	0.513	0.523	0.105	0.498
	OE	0.982	0.861	0.975	0.985	0.766	0.972	0.970	0.627	0.950
Unknown	OS	0.604	0.090	0.573	0.522	0.030	0.496	0.596		0.596
	OE	0.977	0.739	0.962	0.980	0.617	0.961	0.960		0.960
Misc.	OS	0.604	0.144	0.577	0.529	0.044	0.514	0.578		0.578
	OE	0.963	0.700	0.947	0.979	0.600	0.967	0.959		0.959

***Figure C.2*. Average measured OS and OE scores by series, category, and label for the highest $K_{I_{RGW}}$ selection criterion.**

Highest pass_pred selection										
		Household			Tool			Component		
		Pass	Fail	Average	Pass	Fail	Average	Pass	Fail	Average
Known	OS	0.682	0.090	0.634	0.592	0.018	0.558	0.580	0.151	0.571
	OE	0.984	0.596	0.953	0.965	0.624	0.945	0.937	0.705	0.932
Unknown	OS	0.618	0.051	0.595	0.559	0.107	0.541	0.590	0.106	0.577
	OE	0.978	0.725	0.968	0.960	0.668	0.948	0.954	0.749	0.949
Misc.	OS	0.628	0.143	0.592	0.571		0.571	0.556	0.128	0.546
	OE	0.927	0.749	0.913	0.986		0.986	0.966	0.689	0.959

***Figure C.3*. Average measured OS and OE scores by series, category, and label for the highest $pass_{pred}$ selection criterion.**

Stage 3 selection										
		Household			Tool			Component		
		Pass	Fail	Average	Pass	Fail	Average	Pass	Fail	Average
Known	OS	0.684		0.684	0.589		0.589	0.661	0.035	0.649
	OE	0.986		0.986	0.986		0.986	0.986	0.895	0.985
Unknown	OS	0.630		0.630	0.591	0.065	0.573	0.650		0.650
	OE	0.973		0.973	0.984	0.745	0.976	0.986		0.986
Misc.	OS	0.688	0.181	0.675	0.556		0.556	0.636	0.000	0.632
	OE	0.980	0.870	0.977	0.989		0.989	0.972	0.630	0.970

***Figure C.4*. Average measured OS and OE scores by series, category, and label for the stage 3 selection criterion.**

			Round 2							
			Random selection		Highest KI_RGW		Pass_pred		Stage 3	
Series	Object	Description	Pass	Fail	Pass	Fail	Pass	Fail	Pass	Fail
Known	kH001	Sellotape adhesive tape	10	0	10	0	10	0	10	0
	kH002	Figurine	7	**3**	8	**2**	9	**1**	10	0
	kH003	Pencil sharpener	10	0	10	0	8	**2**	10	0
	kH004	Factis eraser	10	0	9	**1**	9	**1**	10	0
	kH005	Pencil	10	0	10	0	10	0	10	0
	kT001	Electrical switch	9	**1**	8	**2**	8	**2**	10	0
	kT002	Small cross wrench	8	**2**	10	0	10	0	10	0
	kT003	Large hex key	10	0	10	0	10	0	10	0
	kT004	Small adjustable wrench	10	0	9	**1**	9	**1**	10	0
	kT005	Small side cutters	10	0	10	0	10	0	10	0
	kC001	Small silver bolt	10	0	9	**1**	10	0	10	0
	kC002	Rod end bearing	10	0	10	0	9	**1**	10	0
	kC003	Pull start handle	10	0	10	0	10	0	10	0
	kC004	Pneumatic T-splitter	10	0	10	0	10	0	9	**1**
	kC005	XCPC pneumatic valve	6	**4**	8	**2**	10	0	10	0

***Figure C.5. pass/fail* outcome of each known object by selection strategy.**

			Round 2							
			Random selection		Highest KI_RGW		Pass_pred		Stage 3	
Series	Object	Description	Pass	Fail	Pass	Fail	Pass	Fail	Pass	Fail
Unknown	uH001	Bluetooth earphone	8	**2**	10	0	10	0	10	0
	uH002	Black Officemax pen	10	0	10	0	10	0	10	0
	uH003	Small foldback clip	10	0	10	0	10	0	10	0
	uH004	Blue sealer	10	0	10	0	10	0	10	0
	uH005	Bic permanent marker	10	0	8	**2**	10	0	10	0
	uH006	Key	6	**4**	5	**5**	7	**3**	10	0
	uH007	Small blue peg	10	0	10	0	10	0	10	0
	uH008	Pencil refill	10	0	10	0	10	0	10	0
	uH009	Rose nail clipper	10	0	10	0	10	0	10	0
	uH010	Cruze glide USB	10	0	10	0	10	0	10	0
	uH011	Blue dragonfly toy	10	0	8	**2**	8	**2**	10	0
	uH012	Green dart	9	**1**	10	0	10	0	10	0
	uH013	Purple grape highlighter	10	0	10	0	10	0	10	0
	uH014	Blue Plesiosaur toy	10	0	10	0	9	**1**	10	0
	uH015	Light green clothes peg	10	0	10	0	10	0	10	0
	uT001	Size 6 wrench	10	0	10	0	10	0	10	0
	uT002	Steel swivel	8	**2**	10	0	10	0	10	0
	uT003	Blue carabiner	10	0	10	0	10	0	10	0
	uT004	Walkers nutcracker	7	**3**	10	0	9	**1**	10	0
	uT005	Blue and yellow screwdriver	6	**4**	7	**3**	8	**2**	9	**1**
	uT006	Small red and black screwdriver	10	0	10	0	10	0	10	0
	uT007	Green USB converter	10	0	10	0	10	0	10	0
	uT008	Small pliers	9	**1**	10	0	10	0	10	0
	uT009	Red spring clip	10	0	9	**1**	10	0	10	0
	uT010	Black USB converter	10	0	10	0	10	0	10	0
	uT011	Silver flathead screwdriver	10	0	10	0	10	0	10	0
	uT012	Black power jack part	10	0	10	0	10	0	10	0
	uT013	Quarter inch audio Y-splitter	10	0	10	0	10	0	10	0
	uT014	Quarter inch audio T-splitter	10	0	10	0	10	0	10	0
	uT015	Silver combination lock	8	**2**	6	**4**	**7**	**3**	6	**4**
	uC001	Motorcycle chain	5	**5**	10	0	8	**2**	10	0
	uC002	Aluminium corner component	10	0	10	0	10	0	10	0
	uC003	Small black component	10	0	10	0	10	0	10	0
	uC004	Large rubber seal	10	0	10	0	10	0	10	0
	uC005	Silver corner pneumatic attachment	10	0	10	0	10	0	10	0
	uC006	Large pneumatic nut	10	0	10	0	10	0	10	0
	uC007	Shaft bearing	10	0	10	0	8	**2**	10	0
	uC008	Square driver socket component	10	0	10	0	10	0	10	0
	uC009	Large blue nylock nut	10	0	10	0	10	0	10	0
	uC010	Arduino Nano	10	0	10	0	10	0	10	0
	uC011	Circular black servo component	10	0	10	0	10	0	10	0
	uC012	Small D-clamp	9	**1**	10	0	10	0	10	0
	uC013	Silver eyelet screw	8	**2**	10	0	10	0	10	0
	uC014	Black rubber HDMI blank	10	0	10	0	10	0	10	0
	uC015	VGA connector	8	**2**	10	0	10	0	10	0

***Figure C.6. pass/fail* outcome of each unknown object by selection strategy.**

			Round 2							
			Random selection		Highest KI_RGW		Pass_pred		Stage 3	
Series	Object	Description	Pass	Fail	Pass	Fail	Pass	Fail	Pass	Fail
Misc.	mH001	Blue blueberry highlighter	9	**1**	10	0	10	0	10	0
	mH002	Red dart	10	0	10	0	10	0	10	0
	mH003	Blue Captain America figurine	10	0	10	0	10	0	10	0
	mH004	Fixed figurine	8	**2**	8	**2**	7	**3**	9	**1**
	mH005	Black and blue pencil	9	**1**	10	0	10	0	10	0
	mH006	Small green peg	10	0	10	0	10	0	10	0
	mH007	Brown scorpion toy	9	**1**	10	0	9	**1**	10	0
	mH008	Red Euoplocephalus toy	10	0	10	0	10	0	10	0
	mH009	Red sealer	10	0	10	0	10	0	10	0
	mH010	PK chewing gum	6	**4**	5	**5**	5	**5**	8	**2**
	mH011	Unopened needle	9	**1**	10	0	10	0	10	0
	mH012	Small black key lock	10	0	10	0	10	0	10	0
	mT001	Medium hex key	10	0	10	0	10	0	10	0
	mT002	Small nail clipper	10	0	10	0	10	0	10	0
	mT003	Wooden handle	10	0	10	0	10	0	10	0
	mT004	Red carabiner	9	**1**	10	0	10	0	10	0
	mT005	Blue Officemax pen	10	0	9	**1**	10	0	10	0
	mT006	Green Officemax pen	10	0	10	0	10	0	10	0
	mT007	Red Sharpie	10	0	10	0	10	0	10	0
	mT008	Golden pen	9	**1**	9	**1**	10	0	10	0
	mT009	Small red screwdriver	9	**1**	9	**1**	10	0	10	0
	mT010	Large red screwdriver	10	0	10	0	10	0	10	0
	mC001	Black power jack	10	0	10	0	10	0	10	0
	mC002	Grey USB cover	9	**1**	10	0	9	**1**	10	0
	mC003	Small black rubber insert	10	0	10	0	10	0	10	0
	mC004	Silver battery terminal	10	0	10	0	9	**1**	9	**1**
	mC005	Small nylock nut	10	0	10	0	10	0	10	0
	mC006	Grey 3D printed part	10	0	10	0	10	0	10	0
	mC007	1oz sinker	9	**1**	10	0	9	**1**	10	0
	mC008	XCPC pneumatic valve – no fittings	7	**3**	10	0	9	**1**	10	0
	mC009	Black pneumatic line	10	0	10	0	10	0	10	0
	mC010	Red resistor	9	**1**	10	0	10	0	10	0
	mC011	Water tap filter	10	0	10	0	10	0	10	0
	mC012	Green terminal black	10	0	10	0	10	0	10	0
	mC013	Door handle	10	0	10	0	10	0	10	0
	mC014	Ball bearing	9	**1**	10	0	10	0	10	0
	mC015	Long black bolt	10	0	10	0	10	0	10	0
	mC016	Steel U	10	0	10	0	10	0	10	0
	mC017	Black circular motor coupling	10	0	10	0	10	0	10	0
	mC018	Power jack with wire	9	**1**	10	0	10	0	10	0

***Figure C.7. pass/fail* outcome of each misc. object by selection strategy.**

Appendix D Glossary

Softmax	A softmax function is often used in the last activation of a neural network to normalise the output to a probability distribution—providing a metric that could be interpreted as a confidence level associated with the classified sample.
Epoch	An epoch refers to the number of passes through a training set
Mini-batch	Mini-batches are grouped clusters of training samples. Generally, it is desirable to process small subsets of training data, as opposed to sweeping through entire training set in one iteration. This allows for the model weights to be re-assessed with each mini-batch.
Iteration	An iteration refers to the number of times the weights of a learning algorithm are updated. If mini-batch training is used, one iteration refers to the number of batches assessed by the model.