DEVELOPMENT OF A NOVEL METHODOLOGY FOR THE MEASUREMENT OF RED19 KIWIFRUIT COLOUR QUALITY

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

Master of Engineering

In

Mechatronics

SAMUEL A. LOCKWOOD-GECK

February 2021

Abstract

Consumers love visually appealing fruit. Colour is a key aspect of this appeal and is determined by a multitude of physical and psychophysical phenomena. Compared to its green and gold counterparts, the colour of red kiwifruit is not easily measured using existing techniques due to the spatially varying nature of its flesh colour. Red19 kiwifruit's flesh colour often presents as a mixture of green, yellow, and red to varying degrees. The current method used to measure kiwifruit redness is by a subjective observer, but this produces a noisy and unreliable dataset which makes quality control difficult.

The objective of this thesis is to create a colour measurement system for Red19 kiwifruit that transforms the current qualitative system into a quantitative system that can be utilised by the kiwifruit industry to produce standard and reliable measures of kiwifruit redness.

Considering this goal, this thesis aims to explore what constitutes an objective grading scale of Red19 kiwifruit, and how a colour measurement system might be created that is invariant to changes in illuminant and spectral sensitivity (observers) and how this system could be deployed on a typical consumer smartphone.

This thesis presents a comprehensive exploration of colour constancy in relation to red kiwifruit. A new scale of redness is proposed, and a small sensory trial was undertaken to validate this scale. A dataset of hyperspectral and raw images was collected and utilised to create a model that estimates the average reflectance of a single red kiwifruit. Another model was then created to regress from the average reflectance of a kiwifruit to a final redness score where fruit were graded based on the outcome of the sensory trial.

The new redness scale proposed therein spans a range of 0 to 10,000 and is calculated by taking the difference of a kiwifruit's average red and green RGB channel in the AdobeRGB colour space. This range was also divided by five to produce alternative class-based scale containing five classes each spanning sections of 2000 units over the new redness scale. This discrete scale is similar to the current qualitative scale used by Zespri to assess the colour of red kiwifruit.

Through the application of a convolutional machine learning model, the average reflectance spectra of a kiwifruit can be estimated producing an RMSE of 0.0109 and 0.0096 over the visible and visible NIR spectral ranges. A general regression model is then used to regress from a kiwifruit's average reflectance spectra to a final redness score and this produces a mean absolute error of 213.82 with a standard deviation of 213.82 which is equivalent to an average error of 2.2% and standard deviation of 2.13% when considering the full range of scores on the redness scale. These models are then combined to produce a final model, named KiwiNet, that produces a correct kiwifruit classification rate of 91.06%. This model has been demonstrated to run on a typical consumer smartphone and can produce a reflectance estimate and redness score for each kiwifruit. This model has been demonstrated to be invariant to five different illuminants and twelve different spectral sensitivities.

Future work should look to carry out a larger sensory trial and explicitly corelate the proposed redness scale to the existing one. Likewise, this study highlighted that a kiwifruit image taken at 583nm appears to be used by the model to estimate kiwifruit redness score and work to reconstruct this wavelength from RGB/RAW images could provide a single measure of redness in the future.

Keywords: Kiwifruit, Colour Constancy, CNN, GRNN, Spectral Super-Resolution

Acknowledgements

I would firstly like to thank Zespri for their generous financial support of this project. Likewise, I would like to thank Massey University's School of Food and Advanced Technology and Massey Agrifood Digital Labs for their support and administration throughout this project.

My thanks to my supervisor Steven Le Moan for bearing with my endless stream of questions and pulling me out of more than a few rabbit holes. Likewise, to my supervisor Gabe Redding for his calm and uplifting approach to the project and insightful feedback. It has been a pleasure working with you both.

I would like to thank Google Calendar as this project would have never been completed without it and the supreme coding language MATLAB for its outstanding UI and functionality.

A final thanks to my flatmates, friends, and family for their support and patience throughout the project.

Table of Contents

Table of Figures

List of Tables

Table of Equations

List of Abbreviations

Abbreviation	Meaning
CAM	Class Activation Mapping
Conv	Convolution
CIE	Commission Internationale de l'Éclairage
CTR	Controlled Temperature Room
CCT	Colour Calibration Target
G3	Gold 3
GAN	Generative Adversarial Network
GLOBALG.A.P	Good Agricultural Practices Standard
GRNN	General Regression Neural Network
HS	Hyperspectral
HRNet	Hierarchical Regression Network
HSC	Hyperspectral Camera
HSI	Hyperspectral Image
ICVL	Imperial Computer Vision Lab
LDA	Linear Discriminant Analysis
MAE	Mean Absolute Error
MSC	Multiplicative Scatter Correction
MSE	Mean Square Error
NFC	Near Field Communication
NIR	Near Infrared
NN	Neural Network
PC	Principal Component
PCA	Principal Component Analysis
RGB	Red, Green, Blue
RMSE	Root Mean Square Error
SAM	Signal Angle Mapper
SGD	Stochastic Gradient Descent
SLR	Single Lens Reflex
SNR	Signal to Noise Ratio
SNV	Standard Normal Variate
sRGB	Standard Red, Green, Blue
SSR	Spectral Super Resolution
SVD	Single Value Decomposition
SVM	Support Vector Machine
SWOP	Yellow, Magenta, Cyan, Black
VIS	Visible Range
wandb	Weights and Biases
YMCK	Yellow, Magenta, Cyan, Key

1.0 Introduction

1.1 Context and Background

Globally, the kiwifruit industry is estimated to be worth more than USD $7.6 billion (GlobeNewswire, 2020). Whilst a large industry globally, Kiwifruit is produced by only a handful of countries such as China, New Zealand, Italy, Greece, and Iran (Statista, 2022).

The New Zealand Kiwifruit industry is no ordinary fruit industry. Shaped by the *Kiwifruit Restructuring Act 1999* and *Kiwifruit Export Regulations Act 1999*, this legislation established Zespri Group Ltd (Zespri) as the single desk seller of New Zealand kiwifruit outside of Australia and New Zealand (New Zealand Kiwifruit Growers Incorporated, 2019). Whilst at times controversial, the structure of the New Zealand kiwifruit industry has enabled the industry to balloon into New Zealand's most valuable horticultural export.

In the 2019/20 season, Zespri spent more than NZD $33 million on research and development with roughly half of this investment going towards the development of new plant cultivars (New Zealand Kiwifruit Growers Incorporated, 2020). As a result of this investment in plant breeding, Zespri is constantly trialling new kiwifruit cultivars to assess their potential not just as a new consumer product, but their response to typical growing conditions, and how they react to handling throughout the supply chain.

New Zealand's Kiwifruit export is worth roughly NZD $2.7 billion as of March 2021 (Ministry for Primary Industries, 2021). The kiwifruit industry itself has seen strong and consistent growth over the past five years with export revenues growing roughly 58% from 2017 to 2021 (Ministry for Primary Industries, 2021). This growth can largely be attributed to the development of the new Gold3 (G3) variety, trademarked SunGold, which has been overwhelmingly well received by consumers in international markets. (Zespri Group Ltd, 2021b).

Export revenues generated from SunGold kiwifruit rose approximately 230% between 2017 and 2021 (Ministry for Primary Industries, 2021). Comparatively, the export revenues of the green Hayward variety remained stable across the same time period. SunGold is an excellent example of how efficient and intensive plant breeding programs can generate not only attractive consumer goods but also significant profits.

Whilst SunGold is set to carry the New Zealand kiwifruit industry into the future, New Zealand may have lost the competitive advantage of the G3 plant variety much sooner than expected. In 2016, SunGold was found to have been unlawfully planted and propagated in China and as of 2021, an estimated $1 billion worth of fruit has been produced without permission (Radio New Zealand, 2021; Clark E., 2021).

Whilst a growing concern for New Zealand kiwifruit growers and Zespri alike, In December of 2019, after strong consumer demand in its pre-commercial trials, Zespri announced it would commercialise a new red variety of kiwifruit named Red19, trademarked RubyRed, that offers a similar unique and desirable fruit like that of SunGold (Zespri Group Ltd, 2019).

It is, therefore, imperative that Zespri commercialises RubyRed quickly in order to retain market share and maintain profits for their growers. However, in initial market testing in China there were reports across Chinese social media that the red colour depicted on Zespri's marketing material did not align with the redness of the consumers' physical kiwifruit. This backlash has led Zespri to rebrand their red kiwifruit packaging and invest into better understanding and measuring the redness of RubyRed kiwifruit hence the inception of this project.

1.2 Problem Statement

Zespri's uniquely branded kiwifruit is known for its high quality. This high quality is desirable to consumers and allows Zespri to attain high profit margins from the sale of Kiwifruit. However, establishing a brand to be synonymous with high quality products is not a trivial task. It requires stringent quality assurance programs such as, in Zespri's case, compliance with good agricultural practises (GLOBALG.A.P), crop protection programs, pest management programs, chemical residue assurance, and rigorous clearance criteria that fruit must meet prior to their export to international markets (Zespri Group Limited, 2022). One of these clearance criteria for the SunGold variety is a colour threshold measurement whereby a sample of fruit is measured with a colourimetric instrument to ensure the hue (colour) of its flesh falls within a pre-defined range. The hue value of kiwifruit flesh is measured as an angle using standard colourimetric equipment such as a chromameter.

For SunGold fruit, the hue angle of the fruit's flesh must be between 106.4 - 110.7° (CIE standard observer and D65 Illuminant) depending on the time of season for the fruit to be cleared for export (Zespri Global Extension Team, 2021a).

The current method employed to take these measurements is to take a 2mm slice along the equator at opposing sides of each kiwifruit. A chromameter is then used to measure the hue angle at each incision point and an average is taken to measure the hue angle of a single fruit. This testing is carried out by third-party accredited laboratories that follow a standard testing procedure set by Zespri.

This colour measurement method has presumably worked well thus far for SunGold kiwifruit as can be seen in its meteoric rise in revenue over the past five years. However, this method is suited for SunGold kiwifruit because the colour of its flesh is consistent throughout the whole kiwifruit and two single-point measurements provide a reasonable assessment as to the colour of the whole fruit (figure 1.1) (Zespri Group Ltd, 2022).

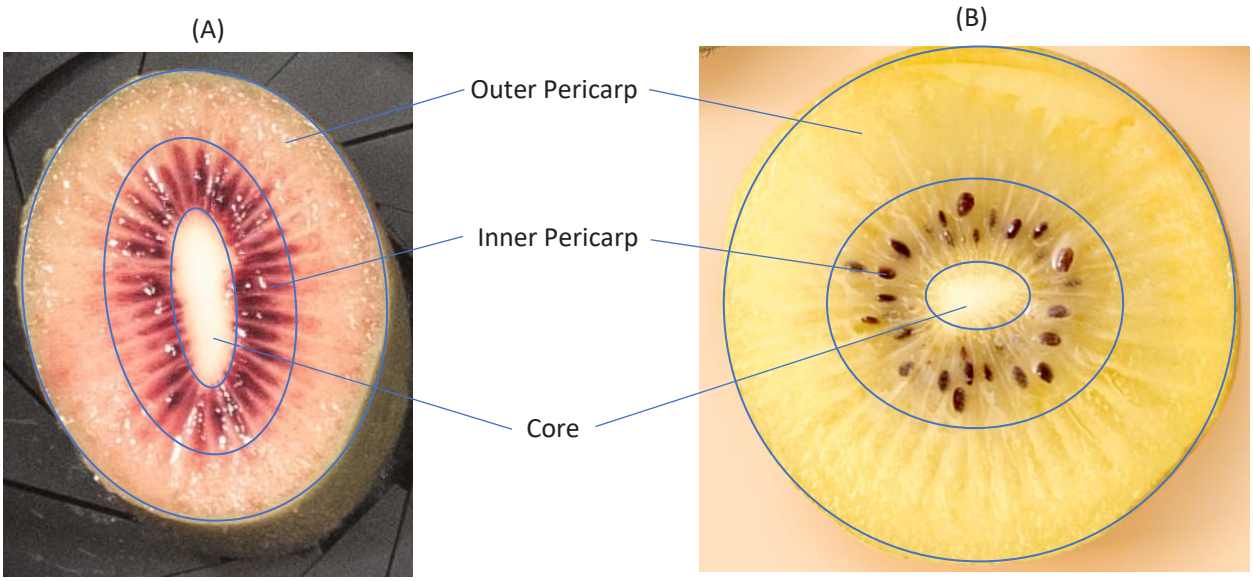


Figure 1.1: Comparison of typical Red19 kiwifruit (A) vs typical SunGold Kiwifruit (B), (Zespri.com).

This is not the case for Red19. Red19 shows a high internal colour variability. Notably, the colour of the inner and outer pericarp differs from bright green to deep red in extreme cases (Figure 1.2)

Figure 1.2: Comparison of extreme colour differences in Red19 Kiwifruit.

Because Red19 presents its internal colour in such a variable way, the current system used to measure SunGold does not sufficiently account for the variability in the spatial dimension of the fruits colour, and thus cannot readily be applied to Red19.

As of the writing of this thesis, the current system implemented to assess the redness of kiwifruit is by way of a subjective observer. A scorer slices a kiwifruit along its equator and visually inspects the internal fruit colour of the cross section. Each kiwifruit cross section is compared to three measures of redness on a psychical scoring guide.

1. Redness spread
   Defined as how well the colour red spreads from the inner to the outer pericarp of the kiwifruit

2. Inner Pericarp Redness
   Defined as the redness of only the inner pericarp

3. Outer Pericarp Redness
   Defined as the redness of only the outer pericarp

This subjective approach has been found by Zespri to produce noisy and inconsistent data which makes the quality control of red kiwifruit difficult. This is especially troubling in Zespri's case, as its brand is synonymous with high-quality kiwifruit, and Zespri leverages this brand to make larger margins on kiwifruit than other kiwifruit sellers.

This then begs the questions: Can a method be developed which is more reproducible and reliable than the current subjective measure? And can this method be deployed cost effectively at scale for use across the industry, from grower, through packhouse, all the way to the end consumer?

1.3 Research Objectives & Questions

This project aims to explore, identify, and assess a method for predicting the colour quality of Red19 kiwifruit. More so, given the current prevalence and ubiquity of smartphones and their recent advances in imaging and computing capabilities, this project will explore the practicality of implementing this method on a typical modern smartphone.

This method will consist of capturing an image of a kiwifruit and then producing a consistent and reliable prediction of fruit redness independent of the type/model of smartphone and the lighting conditions in which the image is captured.

This thesis aims to answer the following research questions:

1. What constitutes a relevant objective grading scale for Red19 kiwifruit redness?
2. Can we create a colour measurement system that is invariant to changes in illuminant and spectral sensitivity that can predict the redness of kiwifruit on this scale?
3. How might this be implemented in a practical way?

1.4 Thesis Structure

This thesis consists of six chapters: Introduction, Literature Review, Methodology, Results, Discussion, and Conclusions/Recommendations. The literature review explores the basics of colour, colour constancy using smartphones, and the state-of-the-art in spectral super-resolution and illuminant estimation.

The methodology outlines the processes used throughout this project to capture data and ensure it is of high quality. The results show the outcome of the experiments that were used to create a scale of kiwifruit redness and develop a unique colour measurement system for measuring this scale that is invariant to change in illumination and spectral sensitivity.

The discussion explores what these results mean, their implications for red kiwifruit colour measurement, areas of potential future research, and the limitations of this study. Finally, the conclusion summarises the findings and implications of this body of research as a whole.

2.0 Literature Review

This chapter delves first into the basics of colour and the methods used to emulate colour in print and digital media. Next, this chapter explores what instruments are used to measure reflectance and predict colour in different environments, and how reflectance has previously been measured using a smartphone. Finally, this chapter explores the state-of-the-art in spectral super-resolution (SSR) and illuminant estimation as a means of addressing invariance to spectral sensitivity and illuminance.

2.1 Beauty is in the eye of the beholder | The Basics of Colour

Lime green, sky blue, and hazelnut brown. Such terms are commonplace in our everyday lives. Yet, we seldom stop to consider the deeper meanings of our daily descriptions of colour. Most people are able to identify whether a feather is heavier than a brick, one is one gram and the other, one thousand. But if asked which colour is more red - Apple red or Ferrari red? A similar quantification is much less obvious. This is because colour, much like taste, smell, hearing, and touch, is a subjective sensation which depends on both physical and psychological factors (Ohta & Robertson, 2005; Fairchild, 2014).

2.1.1 Understanding the Human Visual System

Colour vision, as we experience it every day, begins when photons, the quantum particles emitted by light sources and reflected by physical objects, interact with photosensitive material in the eye. A highly specialised process converts the signal of a light-activated visual pigment (neuron) within the eye into an electrical signal or impulse that is then interpreted by the brain; this process is known as photo-transduction (Hunt & Carvalho, 2016).

In the case of human vision, there are two primary types of neurons in the eye that are sensitive to light. These are referred to as Rods and Cones. Cones are primarily responsible for photopic vision (the perception of colour in relatively bright environments like during the day), whereas Rods provide humans with scotopic vision (colourless vision in low light conditions like at night) (Ohta & Robertson, 2005).

Cones are responsible for colour perception. There are three types of cones in the eye that responds to short, medium, and long wavelengths of visible light (between 400 and 700nm) that, at a basic level, are often considered to perceive blue (short), green (medium), and red (long) wavelengths of light. The peak sensitivity of the short, medium, and long cone cells is 430nm, 530nm, and 560nm respectively (Hunt & Carvalho, 2016; Vienot & Valraven, 2007).

The presence of three photoreceptors with different sensitivities to light allows the human visual system to perceive colour. It is not strictly the absolute value of these signals, but rather the differences between them that our visual system uses to perceive colour (Hunt & Carvalho, 2016); this is referred to as cone opponency (Lee & Silveira, 2016) and is illustrated in Figure 2.2.

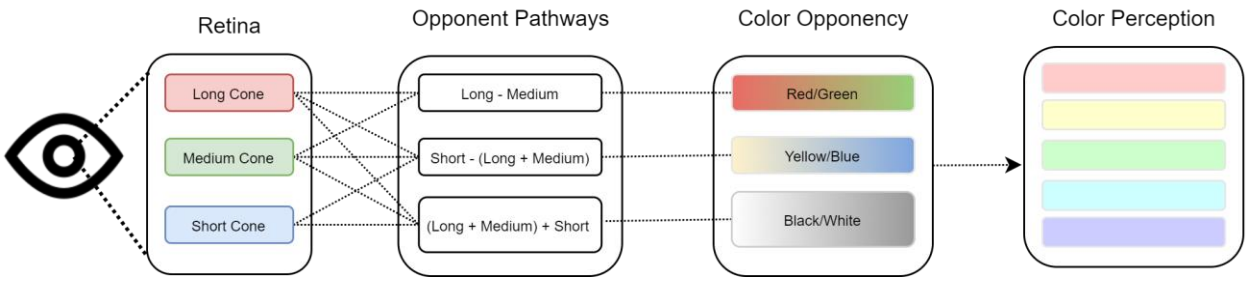


Figure 2.1: Diagram of cone opponency process where the excitation of rod and cones in the retina are encoded along different opponent pathways to produce opponent colours and the perception of colour (Lee & Silveira, 2016).

Another complex natural phenomenon in colour perception is referred to as metamerism. Two objects are considered to be metameric when they appear the same colour under one illuminant and a different colour under another (Figure 2.3). That is, the two objects are of matching colour under a given illuminant but have different spectral compositions (Hunt & Pointer, 2011).

Figure 2.3 highlights how a change in illuminant can make two objects which initially look the same colour, appear different colours when their illuminant is changed. This can also be caused by change in spectral sensitivity of the observer not just changes in illuminant (Hunt & Pointer, 2011). Metamers are defined as a set or pair of colour stimuli that produce the same tristimulus values, such as the same RGB values, when computed using the standard colourimetric observer (Hunt & Pointer, 2011).

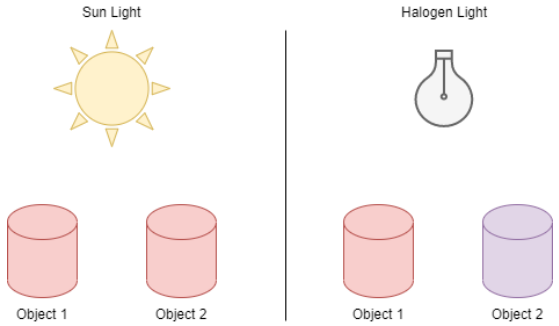


Figure 2.2: Diagram showing how metamerism can make objects appear the same colour or different colour depending on the light source.

Not only does the human visual system actively respond to changes in illumination, it also actively responds to changes in the colour or spectra of the illuminant through a process called chromatic adaption. Chromatic adaption is a process that occurs when we experience a change of illumination e.g., from sunlight to incandescent light, the colour produced by an object largely remains the same (Ohta & Robertson, 2005).

This phenomenon is also referred to as colour constancy, although colour constancy also incorporates the role of memory and expectation/prior learning into the perception of colour (Olkkonen & Ekroll, 2016; Ohta & Robertson, 2005; Hunt & Pointer, 2011). Chromatic adaption works by varying the sensitivity of the three types of cone cells in the eye so that the final or resultant colour of the object closely resembles that of its daylight conditions (Ohta & Robertson, 2005), although there are differing theories as to how colour constancy is achieved by the human visual system.

There exist other observable phenomena produced by the human visual system that relate to spatial features within a scene such as simultaneous contrast and spreading (Johnson & Fairchild, 2003). Similarly, luminance

phenomena such as lightness crispening and chroma crispening make the same colours appear to differ significantly depending on the brightness or chroma of the background (Johnson & Fairchild, 2003). Other phenomena such as the Hunt effect, Steven effect, Helmholtz–Kohlrausch effect, and Bezold–Brücke hue shift all highlight the unique and complex way the human visual system operates (Johnson & Fairchild, 2003).

There is still much to be discovered about the human visual system and how we perceive colour. Work continues to explore what physical properties correlate to their perceptual colour representations, how and if colour can be described independently of environmental factors, and how best to describe and measure how people experience colour (Olkkonen & Ekroll, 2016). However, although colour constancy is a *large* and evolving field, scientists have developed sound ways to effectively "measure" and reproduce colour that occur all around us in our everyday lives.

2.1.2 Mimicking Tristimulus Vision

Cameras closely resemble the human eye in function, this is by design (Figure 2.4) (Ohta & Robertson, 2005). Cameras contain components for controlling the amount of light entering the camera (pupil), a lens for focusing light (cornea), and a photodetector for translating light into electrical signals (retina).

Like the human eye, most cameras contain a means of measuring more than two different wavelengths of light. This is typically achieved using a colour filter array (CFA) (although this is not always the case) that allows the camera sensor to measure three different wavelengths of light, and thus allows the camera to 'perceive' tristimulus (sensitivity to three different ranges of wavelengths) colour.

However, if the spectral response of the camera sensor does not then match that of the human eye, it is unlikely that it will reproduce what is considered as a perceptually accurate colour image. This means that unless a camera's spectral responsivity (its photosensitivity as a function of wavelength) is known, it will be not able to reproduce perceptually accurate colours (Fairchild, 2014).

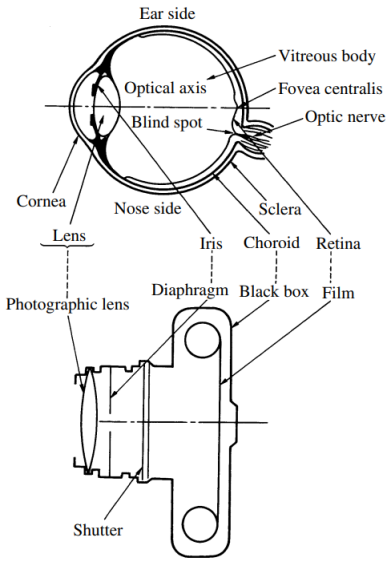


Figure 2.3: Comparison of the human eye and a key components of a typical camera reprinted with permission (Ohta & Robertson, 2005) © 2005 John Wiley & Sons.

The *Commission Internationale de l'Éclairage* (CIE) has produced a standard set of spectral responsivities for the average human eye (International Commission on Illumination, 2019-06) which is widely used in colorimetry research. The CIE also recommends a series of functions for modelling processes such as chromatic adaption that can estimate how human vision adapts to different scenes under differing illumination.

For a camera to reproduce a perceptually accurate colour image, it does not need to have a spectral response that exactly replicates the CIE responses of the human eye (Fairchild, 2014). Instead, a camera needs at least three spectral channels with responses that are a linear transformation of the human cone responses (Figure 2.5) (Fairchild, 2014). This criterion is known as the Luther condition (Luther, 1927) or Maxwell-Ives criterion (Ives, 1990).

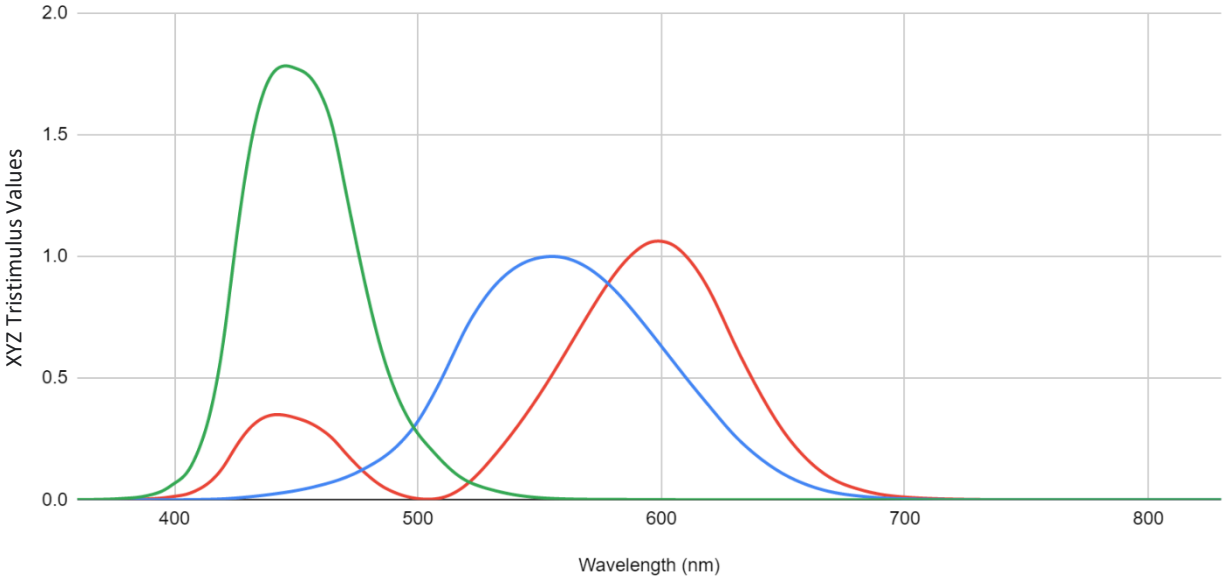


Figure 2.4: Average spectral sensitivities of the human eye defined by the standard colorimetric observer 1931 in XYZ colour space created from data provided by (CIE, 2018)

However, manufacturing a photodetector that has a perfectly linear response is challenging with our current technology. In a comparison of two single lens reflex (SLR) cameras, Fairchild (2014) highlights, by measuring the spectral responsivity of each camera, that the two cameras mimic the sensation of colour differently, both compared to each other and compared to the standard human observer – the cameras 'see' colour differently.

2.1.3 Digital Colour Reproduction

There is much to be considered in photography such as the physical camera components, the image capture settings, and the post-processing operations that work to create a colour image.

At the core of every camera is a sensor that converts light into an electrical signal (Parulski & Spaulding, 2003). Photodetector sensors are usually either charged coupled devices (CCD) or complimentary metal-oxide semiconductors (CMOS) (Parulski & Spaulding, 2003). The main difference between these sensors is the use of a photocapacitor (CCD) or photodiode (CMOS) as the light-sensitive element and this gives CCD and CMOS sensors distinctly different spectral responsivities (Parulski & Spaulding, 2003). CCD devices are generally considered to have superior photo quality compared to CMOS sensors although modern CMOS sensors are fast approaching the same level of performance (Mehta et al., 2015).

CMOS sensors are almost always deployed on smartphones due to their low cost of manufacture (Mehta, Patel, & J. Mehta, 2015), favourable power consumption, and easy integration into microelectronics (Parulski & Spaulding, 2003). The latest generation of the most prevalent smartphones such as iPhone and Samsung Galaxy devices all use CMOS sensors in their cameras (Tominaga et al., 2021). As such, preference will be given to CMOS sensors for the proceeding review.

Cameras achieve tristimulus vision using filters of different colours which are placed in front of the camera sensor within the device. The spatial correlation between pixels, where neighbouring pixels naturally contain information related to surrounding pixels, is exploited by means of a CFA which allows different wavelengths of light to fall onto differing spatial regions of the sensor (Parulski & Spaulding, 2003). A typical CFA is the RGB Bayer pattern array (Figure 2.6) which comprises of a chequered pattern that allows for red, green, and blue light to pass through it and fall onto the camera sensor. This array has two green channels for every one red and blue channel (Parulski & Spaulding, 2003). The doubling of green information allows for improved perceived sharpness of the resultant image as humans determine luminance information primarily from green light (Parulski & Spaulding, 2003).

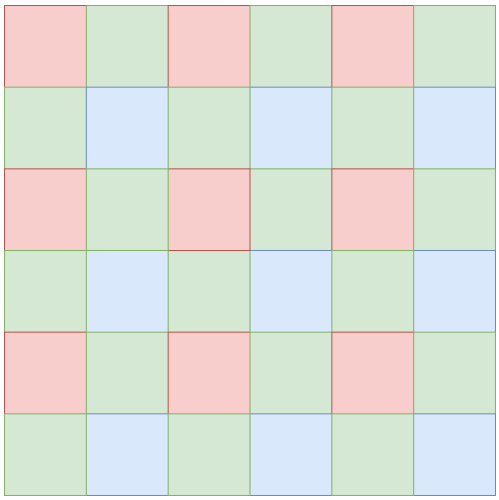


Figure 2.5: Typical RGGB Bayer Patter

This CFA pattern provides the sensors with three channels of light at the expense of spatial resolution. This leaves half of the image where there is no information regarding green light and three quarters of the image contains no information regarding blue or red light. The process of interpolating this chequered colourmetric data from the CFA image into a complete image is known as de-mosaicing (Parulski & Spaulding, 2003).

The process of de-mosaicing images ranges from relatively basic nonadaptive, bilinear interpolation through to dynamic vertical and/or horizontal interpolation filters (Parulski & Spaulding, 2003), and recently the use of neural networks and machine learning approaches (Kwan et al., 2019).

Once de-mosaiced, an image typically needs its white balance adjusted in order to make the image appear perceptually accurate (Parulski & Spaulding, 2003). White balancing refers to adjusting the image to correct for the colour temperature of the image's illuminant (Parulski & Spaulding, 2003). In digital cameras, this is achieved by adjusting the digital values of the image (Parulski & Spaulding, 2003). Many methods for white balancing images exist such as Gray World Theory, its many variations, and Retinex Theory (Zapryanov, Ivanova, & Nikolova, 2012).

The resultant de-mosaiced and white balanced image is still not ready for display or print as the image is still beholden to the spectral sensitivity of the camera. That is, the image is currently saved in the cameras unique colour space (Fairchild, 2014).

Due to the enormous number of cameras, camera sensors, and processing software available to consumers, methods have been developed to ensure images taken from vastly differing devices can be saved, shared, and displayed in reliable and consistent ways. In digital photography, this is achieved by creating a mapping from a camera's device-dependent colour space into an intermediary device-independent colour space (Balasubramanian, 2003). This device-independent colour space is typically a standard set by the CIE such as CIE XYZ or CIE LAB (see section 2.1.4) (Balasubramanian, 2003).

2.1.4 Colour Spaces

The mapping of different cameras' images to a device-independent colour space is important when standardising the reproduction of images. Device independent-colour spaces enable the seamless display of images onto screens and paper although they are not without some trade-offs.

Alongside the mapping of the standard human observer, this information has enabled scientists to produce chromaticity diagrams that show the full range of colour that is perceivable with human vision (Figure 2.7).

However, many displays and printers cannot produce the full range of these colours and standards such as *standard Red, Green, Blue* (sRGB), *Cyan, Magenta, Yellow, Black* (SWOP), and *Cyan, Magenta, Yellow, Key* (CMYK) have all be designed to standardise the range of colours that a screen or page will display (Balasubramanian, 2003).

Considering figure 2.7, the sRGB colour gamut (the area enclosed by its three most extreme values) is a comparatively small region of the chromaticity diagram. The colour gamut itself contains all the possible colours within a given colour space that can be displayed.

AdobeRGB contains a significantly larger gamut than that of sRGB and so it contains literally more colours that can potentially be displayed. sRGB is the international standard for most digital displays (Hunt & Pointer, 2011).

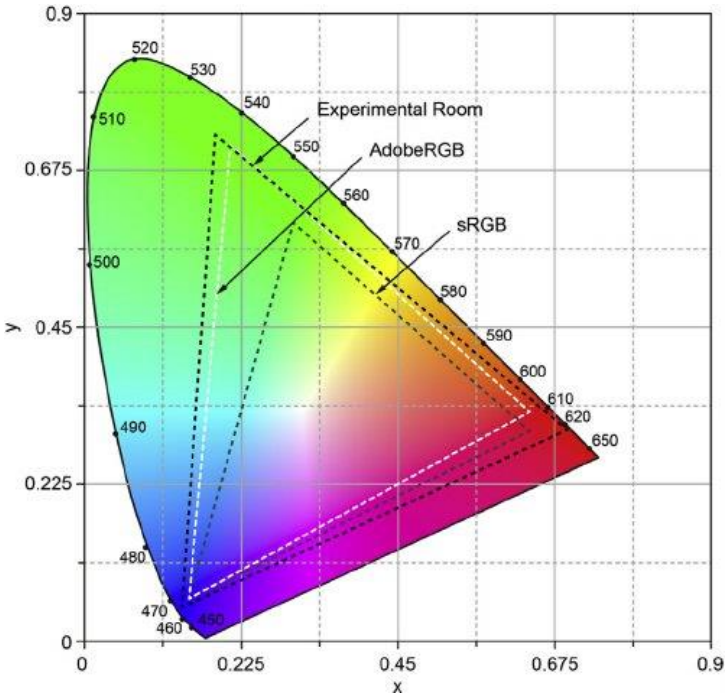


Figure 2.6: CIE chromaticity diagram in XYZ colour space for the standard human observer reprinted with permission from (Choi, Lee, & Suk, 2016) © The Optical Society

The following sections outline how to map images from one colour space to another

2.1.4.1 CIE XYZ

In 1931, the CIE developed a standard system for representing tristimulus values that made colour communication much simpler (Hunt & Pointer, 2011). Whilst the development of the standard colorimetric observer (Figure 2.8) was a large step for colorimetry research, in so far as colour measurements could now be calculated, these values were, at times, confusing due to the presence of both positive and negative values (Hunt & Pointer, 2011).

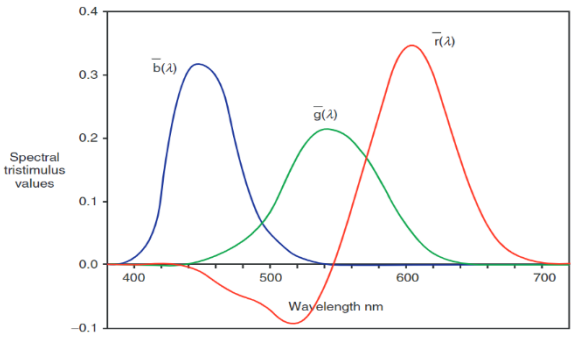


Figure 2.7: CIE Standard Colorimetric Observer, reprinted with permission (Hunt & Pointer, 2011) © 2011, John Wiley & Sons, Ltd

This advancement meant that the luminance of *any* colour could be expressed as a combination of red, green, and blue tristimulus values as per the equation below.

$$L = 1.000R + 4.5907G + 0.0601B$$

Equation 2.1: RGB formula for standard colorimetric observer

The CIE XYZ colour space was created to provide a simpler approach to colour by mapping CIE RGB tristimulus values into the new CIE XYZ tristimulus values (Hunt & Pointer, 2011).The following equations were developed to ensure all colours could be expressed with positive values.

$$X = 0.49R + 0.31G + 0.20B$$

$$Y = 0.17697R + 0.81240G + 0.01063B$$

$$Z = 0.00R + 0.01G + 0.99B$$

Equation 2.2: Formulas to calculate XYZ values from RGB for the standard colorimetric observer

The reason the coefficients of Y are given to five decimal places is so that they occur in the exact same ratio as the luminances used in the CIE RGB tristimulus equation. As such, both coefficients of equation 2.1 and Y of equation 2.2 occur in the same ratio (Hunt & Pointer, 2011) such that,

$$0.17679 : 0.81240 : 0.01063 = 1.0000 : 4.5907 : 0.0601$$

Equation 2.3: Ratio of XYZ Y values vs coefficients from equation 2.1

This characteristic means that Y is proportional to L and thus holds that the ratio of two different colours Y_1, Y_2 coefficients is also equivalent to the ratio of their luminances L_1, L_2 (Hunt & Carvalho, 2016).

2.1.4.2 CIE LUV

Chromaticity diagrams map the relationship between different colour stimuli. However, all colour spaces suffer from some level of non-uniformity. That is, distances in colour that are mathematically equal are not perceptually equal (Hunt & Pointer, 2011).

Early colour spaces such as CIE x,y and CIE x10y10 suffer from a strong non-uniformity when considering the perceptual colour difference of similar mathematical differences (Hunt & Pointer, 2011). For example, the distance that denotes a perceptual change of green colours on a chromaticity diagram is, in some cases, much longer than that of changes in blue colour, see figure 2.9 (Hunt & Pointer, 2011).

While this non-uniformity cannot be completely removed (no colour space is perfectly uniform), the CIE developed the CIE 1971 u' v' chromaticity diagrams that significantly improves this non-uniformity of the colour space.

$$u' = \frac{4X}{(X + 15Y + 3Z)}$$

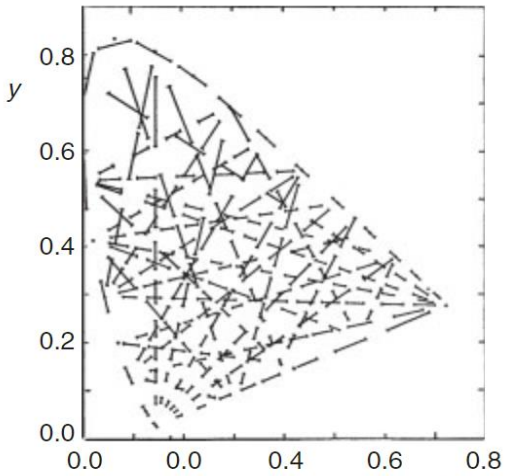


Figure 2.8: CIE x, y chromaticity diagram showing a set of colour difference lines, reprinted with permission (Hunt & Pointer, 2011) © 2011, John Wiley & Sons, Ltd

$$v' = \frac{9Y}{(X + 15Y + 3Z)}$$

Equation 2.4: Calculating u' and v' from XYZ

As shown in figure 2.10, the relative changes in perceptual colour differences are now much more uniform change on the chromaticity diagram (Hunt & Pointer, 2011).

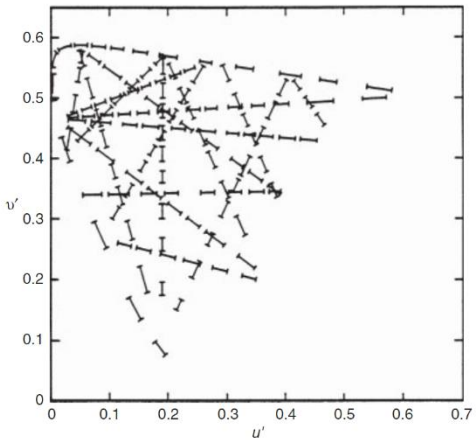


Figure 2.9: CIE u' v' Chromaticity Diagram showing a set of colour difference lines, reprinted with permission (Hunt & Pointer, 2011) © 2011, John Wiley & Sons, Ltd

This then led to the development of the CIE LUV ($L^* u^* v^*$) colour space by considering lightness L alongside the u', v' equations such that,

$$L^* = 116 f(\frac{Y}{Yn}) - 16$$

$$u^* = 13 L^* (u' - u'_n)$$

$$v^* = 13 L^* (v' - v'_n)$$

Where: $f(\frac{Y}{Yn}) = \sqrt[3]{(\frac{Y}{Yn})}$ for $\left(\frac{Y}{Yn}\right) > (\frac{6}{29})^3$

$$f\left(\frac{Y}{Yn}\right) = \frac{841}{108} * \left(\frac{Y}{Yn}\right) + \frac{4}{29} \quad \text{for } \left(\frac{Y}{Yn}\right) \leq (\frac{6}{29})^3$$

Equation 2.5: Calculating Luv from u' and v'

And u'_n, v'_n, Yn are values of u', v', and Yn from a white reference measurement (Hunt & Pointer, 2011).

2.1.4.3 CIE LAB

The CIELAB colour space is another common colour space and is produced by the following equations:

$$L^* = 116 f(\frac{Y}{Yn}) - 16$$

$$a^* = \frac{500}{[f(\frac{X}{X_N}) - f\left(\frac{Y}{Y_N}\right)]}$$

$$b^* = \frac{200}{[f(\frac{Y}{Y_N}) - f\left(\frac{Z}{Z_N}\right)]}$$

Where $f(\frac{X}{Xn}) = \sqrt[3]{(\frac{X}{Xn})}$ for $\left(\frac{X}{Xn}\right) > (\frac{6}{29})^3$

$$f(\frac{X}{Xn}) = \frac{841}{108} * \left(\frac{X}{Xn}\right) + \frac{4}{29} \quad \text{for } \left(\frac{Y}{Yn}\right) \leq (\frac{6}{29})^3$$

And $f(\frac{Y}{Yn}) = \sqrt[3]{(\frac{Y}{Yn})}$ for $\left(\frac{Y}{Yn}\right) > (\frac{6}{29})^3$

$f\left(\frac{Y}{Yn}\right) = \frac{841}{108} * \left(\frac{Y}{Yn}\right) + \frac{4}{29}$ for $\left(\frac{Y}{Yn}\right) \leq (\frac{6}{29})^3$

And $f(\frac{Z}{Zn}) = \sqrt[3]{(\frac{Z}{Zn})}$ for $\left(\frac{Z}{Zn}\right) > (\frac{6}{29})^3$

$f\left(\frac{Z}{Zn}\right) = \frac{841}{108} * \left(\frac{Z}{Zn}\right) + \frac{4}{29}$ for $\left(\frac{Z}{Zn}\right) \leq (\frac{6}{29})^3$

Equation 2.6: Calculating Lab from XYZ

Where Xn, Yn, Zn are X, Y, Z values from a white reference measurement (Hunt & Pointer, 2011).

Both CIE LUV and CIE LAB colour spaces bring a near-uniformity to colour differences, and this allows for a more understandable way of comparing different colours by way of a colour difference. Colour difference, typically denoted ΔE^*, allows for the difference in two colours to be quantified by a single number and is calculated, in CIELAB space, as the Euclidean distance between to colours such that,

$$\Delta E^* = \sqrt{(\Delta L^{*2} + \Delta a^{*2} + \Delta b^{*2})}$$

Equation 2.7: Calculating delta E in Lab colour space

Where $\Delta L^*, \Delta a^*, \Delta b^*$ are the difference in L^*, a^*, b^* values from the two different colours (Hunt & Carvalho, 2016).

Cugmas & Struc, (2020) suggest that a ΔE of 2.3 is generally accepted as the value in which a colour change is perceivable by a typical human observer. However, Cugmas et al., (2017) state that a ΔE below 4 is indistinguishable to the human eye and that the video and printing industries operate with standard errors of between 5 and 6 (Cugmas, Pernus, & Likar, 2017).

2.1.5 The Mathematics of Colour

The previous sections of this thesis have explored many different aspects and the mechanics that operate to alter the appearance of an object and its perceived colour. In the case of the human eye, the activation of single cone can be expressed as the integral, with respect to wavelength (λ), of the spectral distribution of the illuminant, the reflectance of the object, and the spectral responsivity of the cone cell.

$$\delta_k = \int E(\lambda)\ S(\lambda)\ R_k(\lambda)\ \ d(\lambda)$$

Equation 2.8: Calculating the activation of a human cone cell

Where $S(\lambda) = \frac{I_{ref}}{E(\lambda)}$ = Surface Reflectance Factor or the object

δ_k is the receptor output for a given cone *k*, $R_k(\lambda)$ is receptor sensitivity, I_{ref} is the spectral distribution of the reference light, and $E(\lambda)$ is the spectral distribution of the illuminant light (Stanikunas, Vaitkevicius, & Kulikowski, 2004).

Similarly, this equation can be considered in digital photography where the intensity of a single pixel on a camera sensor can be expressed as the integral, with respect to wavelength (λ), of the imaging optics, the spectral sensitivity of each colour channel, the illuminant, the object's reflectance, and the angle of the illuminant.

$$I_C = k(\gamma)cos\theta_i \int_{\lambda\min}^{\lambda\max} S_c(\lambda)\ L_i(\lambda)\ F(\lambda,\theta)\ d(\lambda)$$

Equation 2.9: Calculating the value of a single RGB pixel

Where c∈{R,G,B}, $S_c(\lambda)$ is the spectral sensitivity of the colour channel, $L_i(\lambda)$ is the spectral distribution of the illuminant, $F(\lambda,\theta)$ is the bi-directional reflectance distribution function (reflectance of the object), θ is the angle of the observer in relation to the illuminant, $k(\gamma)$ is a coefficient that represents the imaging optics, and $\lambda\max\ \&\ \lambda\min$ represent the upper and lower bounds of the spectrum of interest respectively (Akkaynak, et al., 2014).

To create transformations from one colour space to another, researchers typically capture a series of images containing objects of known spectral reflectance/colour ($V^{XYZ}_{ground\ truth}$). A transformation matrix T can then be created to map a device-dependant colour representation V^{RGB}_{linear} to device-independent representation V^{XYZ} by least-squares regression where,

$$V^{XYZ}_{ground\ truth} = T\ V^{RGB}_{linear}$$

Equation 2.10: Calculating a transformation matrix from camera-dependant linear RGB to XYZ colour space

Where $V^{XYZ}_{ground\ truth}$ and V^{RGB}_{linear} represent by a 3 x N matrix where N is the number of calibration patches. For a generic ColorChecker calibration target produced by X-Rite, this would create a 3 x 24 matrix for each colour tile where the true $V^{XYZ}_{ground\ truth}$ values are known.

If the V^{RGB}_{linear} image uses the same illuminant as the $V^{XYZ}_{ground\ truth}$ values, a direct transformation can be found. However, this is almost never the case and as such, a white point conversion is needed (Akhtar & Mian, 2018). To adjust the white point of an image a 3x3 matrix from white point S to white point D is needed (Akkaynak, et al., 2014).

$$V^{XYZ}_{dest} = \frac{1}{M_a}\begin{bmatrix} \frac{\rho_D}{\rho_S} & 0 & 0 \\ 0 & \frac{\gamma_D}{\gamma_S} & 0 \\ 0 & 0 & \frac{\beta_D}{\beta_S} \end{bmatrix} [M_a]\ V^{XYZ}_{source}$$

Equation 2.11: Calculating a white point adjustment in XYZ colour space

Where V^{XYZ}_{dest} represents the 3xN matrix of colours in XYZ space for the transformed image, and V^{XYZ}_{source} represents 3xN matrix of colours in XYZ space of the current colour temperature. Where ρ, γ, and β represent the unique tristimulus values for each cone in the eye (Akkaynak, et al., 2014).

$$\begin{bmatrix} \rho \\ \gamma \\ \beta \end{bmatrix} = [M_a][WP]^{XYZ}_i \qquad i = S, D.$$

Equation 2.12: Calculating the response of each cone cell based on the white point of an illuminant

Where WP is a 3x1 vector containing the white point value of each light source. Ideally, the white point of each image is measured using colourimetric equipment, but it is also common to estimate it algorithmically. M_a is a 3x3 matrix which is defined through empirical measurement, the most common being the Von Kries, Bradford, Sharp, and CMCCAT2000 matrices shown below (Susstrunk, Holm, & Finlayson, 2002).

$$M_{Von\ Kries} = \begin{bmatrix} 0.3897 & 0.6890 & -0.0787 \\ -0.2298 & 1.1834 & 0.0464 \\ 0 & 0 & 1 \end{bmatrix}$$

$$M_{Bradford} = \begin{bmatrix} 0.8951 & 0.2664 & -0.1614 \\ -0.7502 & 1.7135 & 0.0367 \\ 0.0389 & -0.0685 & 1.0296 \end{bmatrix}$$

$$M_{Sharp} = \begin{bmatrix} 1.2694 & -0.0988 & -0.1706 \\ -0.8364 & 1.8006 & 0.0357 \\ 0.0297 & -0.0315 & 1.0018 \end{bmatrix}$$

$$M_{CMCCAT} = \begin{bmatrix} 0.7982 & 0.3389 & -0.1371 \\ -0.5918 & 1.5512 & 0.0406 \\ 0.0008 & 0.0239 & 0.9753 \end{bmatrix}$$

With the white point now corrected, the transformation matrix T can be calculated by using the Moore-Penrose pseudo-inverse of the V_{linear}^{RGB} matrix, denoted $\left[V_{linear}^{RGB} \right]^{+}$ by the following equation (Akkaynak, et al., 2014):

$$T = V_{ground\ truth}^{XYZ} \left[V_{linear}^{RGB} \right]^{+}$$

Equation 2.13: Calculating colour transformation matrix using Moore-Penrose pseudo-inverse

The matrix T can then be used on an image to transform it from device-dependent space I_{linear}^{RGB} to device-independent space $I_{corrected}^{XYZ}$ using the transformation matrix T as follows (Akkaynak, et al., 2014):

$$I_{corrected}^{XYZ} = T\ I_{linear}^{RGB}$$

Equation 2.14: Using a colour transformation matrix to represent an image in device-independent XYZ colour space

2.2 Measuring Colour in the field

Special equipment has been developed to accurately measure psychical colour *stimulus* (the signals of light that are part of the perception of colour). It is important to remember that the colour humans perceive does not typically align exactly with the quantifiable physical stimulus that is measured by colorimetry instruments. This is due to the complex and not fully understood processes that occur in the visual system. Namely, tristimulus colorimeters and spectrometers (spectrophotometer or spectroradiometer) are commonly used to obtain pseudo-colour data (Ohta & Robertson, 2005; Ohno, 2007).

Colorimeters are generally considered a quick and simple way to measure pseudo-colour with typical application in production systems (Ohno, 2007). However, due to metamerism, tristimulus colorimeters are not considered suitable in applications that require measurement of different illuminants or objects of vastly different colour (Ohno, 2007). On the other hand, spectrometers, although more expensive, provide a more accurate way of measuring reflectance and do not suffer from many of the shortcomings of colorimeters (Ohno, 2007).

Spectroradiometers typically measure spectral irradiance, spectral radiance, spectral radiant intensity, and total spectral radiant flux (Ohno, 2007). Radiometry refers to the measurement of the radiant power of a stimulus which typically takes interest in the power or energy output of a light source (Hunt & Pointer, 2011). Spectroradiometers also measure the chromaticity of light sources via comparison with a standard power source – usually the CIE standards 2° observer (Ohno, 2007).

Comparatively, spectrophotometers typically measure spectral reflectance, reflectance factor, transmittance, transmittance factor, and bidirectional reflectance distribution function (BRDF) (Ohno, 2007). Photometry refers to the amount of light in a given stimulus (Hunt & Pointer, 2011). Compared to radiometers, spectrophotometers provide insight into the spectral characteristics of the target object.

In short, spectroradiometers are used to measure the spectrum/radiant power of illuminant sources, whereas spectrophotometers are used for capturing the reflectance or transmittance of objects themselves (Ohno, 2007; Hunt & Pointer, 2011).

Current spectrophotometers typically look something analogous to printers from the early 2000's. Bulky and clumsy describe most spectrophotometers available to scientists today. That is not say their colour measurements are lacking. On the contrary, portable spectrophotometers, such as the Konica Minolta CR-400, have a ΔE^*(in the Lab colour space) of 0.6 where a typical human observer will not observe a perceptual change in colour until ΔE^* exceeds 2.3 (Cugmas & Struc, 2020).

Whilst colorimetry refers to the study of light within the visible range (380nm – 780nm), the field of Spectroscopy extends far beyond this range into the near-infrared (NIR) (800nm – 2500nm) and mid-infrared (MIR) (2500nm – 15,000m) spectrum (Manley, 2014).

The NIR spectrum is of great interest to researchers especially when exploring organic materials as it contains information about the major chemical bonds related to the bonding of Hydrogen with Carbon, Oxygen, and Nitrogen (Manley, 2014). This allows researchers to predict properties of materials based on the nuances of their spectra. Hyperspectral cameras (HSC) address the lack of spatial resolution associated with spectrophotometers and allow researchers to capture a three-dimensional dataset featuring two spatial dimensions (x, y) and one spectral dimension (λ) (Figure 2.11), this dataset is often referred to as a datacube or hypercube (ElMasry & Sun, 2010). As such, hyperspectral imaging (HSI) is suitable in imaging applications of highly heterogenous samples (Manley, 2014) like that of Red19.

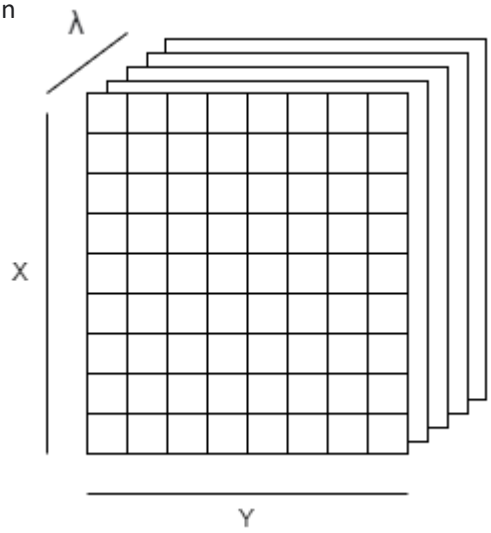


Figure 2.10: Hyperspectral Datacube Structure

The Konica Minolta CR-400 colorimeter is currently used by the kiwifruit industry to measure hue angle and has a spectral responsivity that very closely mimics the human cones response.

This device allows fruit pseudo-colour to be measured reliably, consistently, and accurately although it does trade its very accurate spectral response for extremely low spatial resolution.

Whilst these devices produce these high accuracies by reducing their spatial resolution, it does not appear that at this time a spectrophotometer exists with anywhere near the resolution required to measure the colour across a cross section of a kiwifruit. Figure 2.12 highlights how much colour information is lost when using only a single point reading. It appears that a single kiwifruit image needs to be at least approximately 100x100 pixels to distinguish each region somewhat clearly.

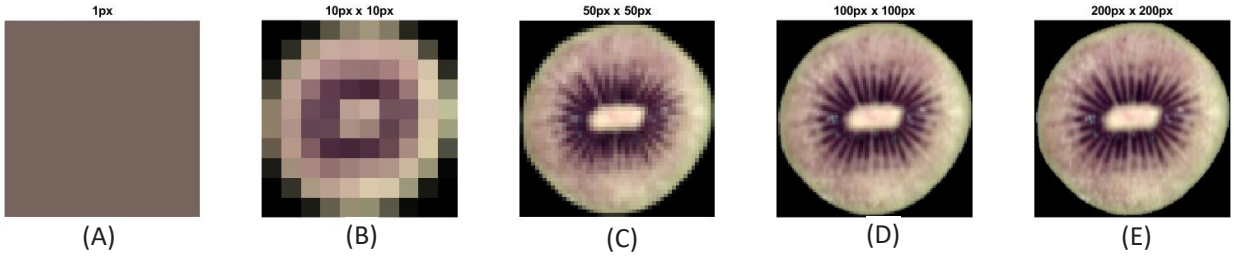


Figure 2.11: Example of kiwifruit images by image size. 1px (a), 10x10px (b), 50x50px (c), 100x100px (d), 200x200px ©

2.3 Smartphone Colorimetry

An increasingly attractive solution to colour measurement arises through the use of smartphones. Smartphones underpin much of our daily lives and, therefore, provide many opportunities to collect valuable data out in the field. Commonly outfitted with high resolution cameras, LEDs, and positioning services, smartphones provide an ideal, low-cost alternative to traditional colour measurement systems (Kılıç, Horzum, & Solmas, 2018).

In recent years, smartphones' colour data has been used across a range of scientific fields including chemical sensing (Li, et al., 2020; Fu, et al., 2019; Garcia, et al., 2011), healthcare (Kap et al., 2021; Karlsen & Dong, 2015; Outlaw et al., 2020; Baherinia & Manduchi, 2011), pH detection (Shen et al., 2012), and food colour analysis (Chen & Fu et al., 2018).

There are many limitations and challenges when using smartphone cameras. Namely, phone cameras are typically optimised for use in environments of high ambient lighting (daytime conditions) (Shen, Hagen, & Papautsky, 2012) and the fluctuations in this ambient lighting (such as the time of day) makes capturing meaningfully comparable images difficult (Shen et al., 2012; Nixon et al., 2019). Likewise, compression algorithms such as JPEG compression quantise colorimetric information (Shen et al., 2012) leading to distortions in colorimetric measurements.

One way to circumvent JPEG compression is by saving images in their RAW format. RAW images are the raw sensor data that has not been compressed or pre-processed (Verhoven, 2010). Put simply, the only processing RAW images have undergone is their digitisation from an analogue electrical signal from the photodiode into a digital one that is stored in the camera's memory (Verhoven, 2010).

Similarly, a typical smartphone camera automatically selects a multitude of settings for the user including shutter speed, ISO, and white balance. Up until recently, these setting were not directly controllable (Kılıç, Horzum, & Solmas, 2018). However, modern smartphones and smartphone applications such as Adobe Lightroom now allow for complete control of camera settings as well as the capture of images in their native RAW format (Nixon et al., 2019). If camera settings can be adequately controlled and RAW images captured,

smartphones have the potential to capture high quality colour images that can be used for colourmetric analysis.

Previous research has explored ways to control the effect of ambient lighting conditions on images using pre-image calibrations and calibration tiles/colour cards (Nixon et al., 2019; Shen et al., 2012) as well as apparatus to completely remove ambient lighting conditions (Mariakakis, et al., 2017; Kılıç, Horzum, & Solmas, 2018; Charriere et al., 2013). Other solutions include flash-no-flash capture methods that allows for the subtraction of ambient lighting post-capture.

Nixon et al (2019) compared three approaches to mapping different smartphone images to the CIEXYZ colour space from a smartphone's device-dependant colour space. One approach uses the image's metadata to map the image to the XYZ space, another uses the open source 'dcraw' software, and the final method uses a colour calibration target (CCT) to explicitly generate a mapping between the two spaces (section 2.1.5).

Nixon et al (2019) found that the dcraw method was not consistent across smartphones and found that a CCT outperformed the metadata approaches in almost all cases. Similarly, Nixon et al (2019) highlight that colour calibrations do not hold across phones even of the same make and model. Each device should be calibrated independently for best results.

Shen et al (2012) established a linear relationship between their custom CCT and colour temperature and used this to create a normalisation algorithm so that images captured under different lighting conditions could be normalised to a standard colour temperature of 5000K before analysis. Shen et al (2012) used this technique to compensate each red, green, and blue set of pixels from captured smartphone images and map them to a standard illuminant before predicting their x and y values. Using this technique, Shen et al (2012) produced nearly identical x,y values when capturing images using a 3500K colour temperature on a system calibrated for measurements at 5000K thanks to this compensation algorithm.

Goni & Salvadori, (2017) found a digital camera system that included a fixed illumination system and CCT produced a ΔE of 5.88 ± 3.22 compared to a colourimeter and that these results produced a noticeable colour difference for most samples. Goni & Salvadori, (2017) highlight that whilst the system does not match the colourimeter in performance, its ability to consider larger spatial information has merit in some applications.

Another use of a CCT was implemented by Wang & Zhang, (2013) to create a colorimetric system for assessing tongue health. Wang & Zhang, (2013) defined a unique colour space specifically for the human tongue and then created a CCT that only featured the unique colours in this space. Wang & Zhang, (2013) showed that this application specific CCT had a 48% lower ΔE compared to using a generic Munsell ColorChecker CCT.

Cugmas & Struc (2020) attached a dermatoscope featuring four LEDs and a polarising filter onto two different smartphones and used both a generic CCT as well as a CCT specifically design for skin tones to calibrate the system. Cugmas & Struc (2020) found that the two smartphones produced a ΔE of 6.45 and 6.60 respectively compared to the ground truth. Similarly, Cugmas & Struc (2020) found that the same phone would produce a ΔE of up to 0.12 on repeat trials in a laboratory setting although in clinical trials this increased to 2.91. Comparatively, different phones produced a ΔE of 3.91 when compared to each other in both laboratory and clinical trials.

Like Wang & Zhang (2013), Cugmas & Struc (2020) found that the use of application specific CCT (in this case a skin tone CCT) significantly decreased ΔE of the system. It is likely that the use of a dermatoscope also

positively impacts colorimetric readings as it provides a decrease in ambient lighting. Charriere et al. (2013) found similar decreases in ΔE using a microscope.

Similar comparisons of smartphone cameras produced ΔE results in the range of 3.5-8.4 (Amani, et al., 2019), 3.–4 - 10.2 (Cugmas et al., 2017), and 3.0-4.4 (Mulcare & Coward, 2019).

Instead of using an intermediary device-independent colour space such as CIE XYZ, Cugmas & Struc (2020) used a CCT to create a mapping directly from each phone's device-independent colour space to CIE LAB colour space. Cugmas & Struc (2020) achieved a ΔE of 3.8 using a first-degree polynomial with four parameters to create this mapping. Cugmas & Struc (2020) also highlight that the application of a quadratic polynomial with ten parameters (based on the work of Zhang et al., (2016)) significantly lowered their ΔE to 0.9.

Moving away from CCTs, Lopez-Ruiz, et al., (2014) implement a colorimetric pH detector through fixed camera parameters, controlled capture conditions, and utilising a smartphone camera's flash for illumination and produced a mean square pH error of 0.167. This system was not tested using different smartphones. Several studies use optical enclosures to completely remove ambient lighting (Garcia, et al., 2011; Szipszky, et al., 2021) and they produce ΔE values as low as 1.59 (Mulcare & Coward, 2019).

Another area of research focuses on supplementing smartphones with custom-made attachments. Fu et al., (2019) create a complimentary ambient light sensor to supplement the estimation of ambient illumination and produced results on-par with those produced using chromatography-mass spectroscopy. Zhang et al., (2016) showed similar success by attaching a microscope to a smartphone to assess stained tissue samples and produced results on-par with benchtop microscopes with a ΔE of 2.02 and 2.36 respectively for different variations of the system. Likewise, Li et al., (2020) developed a UV-vis spectrometer attachment that was able to detect the presence of common heavy metal ions in samples of water when attached to a smartphone camera. Much like optical enclosures, these systems offer significant control over ambient lighting and show low ΔE values as a result.

As per the above review, classical methods of colour constancy typically fall under three key categories: Application of an optical enclosure to control ambient lighting, a CCT to provide a means of calibrating for a scenes illuminant, or by using a standard illuminant and analysing and comparing results only under one illuminant. Whilst these methods produce low ΔE results in some cases, their application to the measurement of red kiwifruit is limited as to implement them at scale will be both costly and time consuming.

In recent years, machine learning has become increasingly popular in most areas of image analysis with colour constancy being one of them. Machine learning applies a less algorithmic approach compared to classical colour constancy techniques, instead opting to utilise a learned/trained model which often leads to state-of-the-art results. However, machine learning does have some draw backs when trying to understand how such methods reach their conclusions and understanding failure cases. This issue, known as the problem of interpretability, is itself the subject of a great deal of research attention (Carvalho, Pereira, & Cardoso, 2019).

2.4 Colour Constancy & Machine Learning

Whilst the application of machine learning approaches in academic research has exploded in recent years, thanks largely to advances in hardware, software, and the availability of large datasets, machine learning is not a new tool in colour constancy. Over 20 years ago, Funt, Cardei, & Barnard, (1998) showed that a neural network outperformed standard illumination correction algorithms such as Gray World, White Patch, and Gamut Mapping by significant margins when estimating chromaticity values from generated scenes. Funt et al., (1998) generated these scenes from a set of captured illuminances and known surface spectral reflectances.

Stanikunas, Vaitkevicius, & Kulikowski, (2004) trained a neural network using nine different illuminants and explored the change in chromaticity (xyY) values when comparing the target's chromaticity values to its background. Stanikunas et al., (2004) found that the network produced low errors when trained with at least five different illuminants and found that the network could not be trained on the target signal alone, suggesting that colour constancy cannot be achieved without the background signal. By varying the background signal that was input into the network and observing its effect on the output values Stanikunas et al., (2004) conclude that colour determination contains two key aspects: the estimation of background colour and the relative difference between the background colour and the target colour.

Popular methods of machine learning methods include support vector machines (SVM) and artificial neural networks – or simply neural networks (NN) (Nitze, Schulthess, & Asche, 2021) although the term machine learning encompasses a far wider area of research including supervised and unsupervised learning, reinforcement learning, and ensemble learning.

Whilst a wide field, SVMs and NNs are the most prominent machine learning techniques in the field of smartphone-based colorimetry and computer vision. SVMs are often used in both linear and non-linear classification and regression problems (Mahesh, 2020) and work by mapping inputs into higher dimensional feature spaces and then identifying boundaries between classes in these high dimensional spaces (Mahesh, 2020).

Comparatively, NNs work by mimicking the human brain through a series of interconnected artificial neurons (Mahesh, 2020). NNs work to achieve their goal by adjusting how each layer of neurons interacts with other layers until an optimal solution is reached through the minimisation of a loss function or output error.

Regarding SVMs, Mutlu, et al., (2017) used a support vector machine (SVM) to classify the colour of pH strips from images taken on a smartphone. Mutlu, et al., (2017) compared both the effect of image type (JPEG and RAW) as well as the effect of illumination on classification accuracy finding that the SVM which used the least squares algorithm produced perfect classification when using the same illuminant both with and without ambient light mitigation. However, Mutlu e al., (2017) found that when using different illuminants, the SVM produced classification accuracies of around 80%. Mutlu, et al., (2017) also found that the image filetype did not have a significant impact on the final result.

Lazaro et al. (2019) developed a near field communication (NFC) based colorimeter that could be used in conjunction with a smartphone and tested a series of classification methods to determine, in a binary way (good or bad), how long a piece of fruit had been outside a controlled temperature environment. Lazaro et al. (2019) compared six classification algorithms and found that for the golden apples and bananas datasets all six classification methods produced equivalent results. Comparatively, nearest neighbour and SVM approaches on the red apple dataset showed the weakest classification accuracy of 47% and 53% respectively.

Extending on the work of Mutlu, et al., (2017), Tania, et al., (2020) compare the effect of six different colour feature descriptors on classification accuracy of pH strips and found that the least squares SVM produced the best classification results compared to six other algorithms thus confirming the findings of Mutlu, et al., (2017).

Kim et al., (2017) compare the performance of a linear discriminant analysis (LDA), SVM, and NN on classifying paper assay colour strips and its reaction with alcohol. This experiment used an optical enclosure as well as a light diffuser and supplementary lens attached to a smartphone to capture images. Kim et al., (2017) tested the effect of using different colour spaces on classification accuracy and found the SVM and NN methods produced similar results across all four colour spaces. Kim et al., (2017) also tested the effect of using different smartphone on classification accuracy and found that the NN outperformed the SVM across five different smartphones.

Similarly, Taneja et al., (2021) found a NN outperformed an SVM when predicting soil organic matter and soil moisture content but only when it was given 22 input features. The NN did not perform as well using only 6 inputs features. Comparatively, Tripathy et al., (2019) found that SVM outperformed a NN when used to classify the pH of milk on an application specific sensor although the NN performed only approximately 1% worse than the SVM. This suggests the use of a SVM or a NN may be largely application specific or that, in some cases, they will produce equivalent results.

Neural network-based colorimetry has been applied widely in chemical and biological sensing. Paper-based assays are commonly used as a means of producing a detectable change of colour that is correlated to either the presence of a specific chemical or the reaction of specific chemicals.

Carrio et al., (2015) used an optical enclosure and test strips to capture images that where then used to train a NN which produced results of between 88-100% classification accuracy. Aznan et al., (2021) also used controlled illumination to capture images of rice and extracted a series of 9 colorimetric features from the resultant images to train a NN and this produced 87.8% classification accuracy.

Although these machine learning approaches produce high classification accuracies in their respective applications, many of the studies mentioned above still use more rudimentary colour constancy techniques such as optical enclosures and fixed illuminants. These capture techniques have shown low ΔE values and high classification accuracies (see section 2.3) without the use of SVMs and NNs for analysis.

Considering the final solution of this thesis may be deployed on a smartphone and used anywhere in the world, colour constancy techniques that require augmentation (CCTs, optical enclosures) are not suited for this purpose. Therefore, that leaves two key areas of colour constancy to explore that do not require these additional psychical augmentations: Illuminant estimation and spectral super-resolution (SSR).

Considering equation 2.9, the signal of any given pixel depends on the reflectance of the target object, the spectrum of the illuminant lighting the scene, and the spectral sensitivity of the camera (Koscevic et al., 2019a; Barron, 2015). Because these three variables are largely unknown for any given scene (with the exception of controlled environments), computational colour constancy is an ill-posed problem and often difficult to solve.

In illuminant estimation, researchers explore ways to estimate the chromaticity of the illuminant which then allows for any scene to be corrected or white balanced to the white point of the illuminant – see section 2.1.5. Comparatively, SSR seeks to recover the reflectance spectra of a scene or objects within a scene. To achieve SSR across a range of illuminances, this SSR model must be invariant to changes in illuminant.

Please note that in the following review the number of layers in a network refers to the number of convolutional (conv) layers, not the total number of layers.

2.5 Illuminant Estimation using Neural Networks

Classical approaches to Illuminant estimation typical fall into one of two categories: statistical-based approaches and physics-based approaches. Grey-World theory, and its many variations, is a common statistical technique that assumes that the average of all reflectances in an image is grey and uses this assumption to white balance the image (Hordley, 2006). Another statistical approach to illuminant is Max RGB where the scene is assumed to contain a surface that is maximally reflective which is then used to adjust the white balance over the whole image (Hordley, 2006). Gamut mapping is another popular statistical technique that estimates a series of possible illuminants that could explain a scenes pseudo-colour and then works to select the most likely illuminant that is then used to white balance the image (Hordley, 2006). Another set of techniques is referred to as Bayesian or probabilistic methods such as Colour by Correlation (CbC). These methods use a statistical prior to determine the likelihood that an image was captured under a certain illuminant, with the illuminant with the highest probability typically being used for white balancing (Hordley, 2006). Comparatively to these statistical approaches, physics-based approaches exploit the interaction of light and surfaces within a scene such as exploiting specular reflections in an image to estimate the illuminant (Hordley, 2006). However, these approaches rely on segmentation of surfaces within the scene which is not trivial and prone to error (Hordley, 2006).

Current state-of-the-art in illuminant estimation is achieved using convolutional neural networks (CNN) (Koscevic et al., 2019a). Authors such as Bianco et al., (2015) trained a five-layer CNN on RAW images to estimate the illumination over a set of regions across an entire image and produced result better than classical methods (Gray World, etc) and existing learning-based approaches. Comparatively, Sulistyo et al., (2016) used a genetic algorithm to learn the transformation from image RGB values to the true MacBeth ColorChecker CCT values by using a set of 24 smaller NNs for each colour tile and then combining the result into a larger network. Sulistyo et al., (2016) showed that their network performed better than classical illuminant estimation methods as well as a NN proposed by Shengxian, et al., (2010).

Shi et al., (2016) propose a unique solution to white balancing by designing two different CNNs: one to produce two estimates for a scene's illuminance, and another to select the most likely true scene illuminance from this set. This network produced state-of-the-art results compared to classical and other learning-based methods when compared using the ColorChecker Dataset (Shi & Funt, 2000). Shi et al., (2016) produced a mean error of 1.90 compared to 7.55 (White-Patch), 6.36 (Grey-World), 4.82 (Bayesian), 4.2 (Pixel-Based Gamut), and 2.36 (CNN) (Bianco et al., 2015). Achieving similar results, Hu et al., (2017) implemented a 7-layer CNN as well as a 25-layer CNN which dynamically segmented and weighted regions of interest within each image and produced illuminance estimation results slightly superior to (Shi et al., 2016).

Implementing larger CNNs generally produced better illumination estimation results as shown by Yuan & Li, (2019) who developed a seven-layer CNN which used the ImageNet (Deng, et al., 2009) architecture in the first five layers to estimate global illuminance of an input image and produced results on-par with other state-of-the-art CNNs such as AlexNet-FC4 (Krizhevsky, Sutskever, & Hilton, 2012), SqueezeNet-FC4 (Hu et al.,2017), and DS-NET (Shi et al., 2016). Similarly, Afifi, (2018) implemented a 9-layer CNN and complimented it with an additional input image that featured segmented regions of interest related to the original RGB image and found that this network produced better illumination correct results compared to classical methods and AlexNet architecture (Krizhevsky et al., 2012).

Like Hu et al., (2017), Koscevic et al., (2019b) proposed a CNN that utilised the pretrained VGG16 (Simonyan & Zisserman, 2014) architecture and also extended the work of Afifi, (2018) by allowing the network to estimate its own regions of interest by adding additional attention mechanics that determined the part of an image that provide the most useful colorimetric information. This network performed better than classical methods although did not perform better than a statistical-based approach proposed by (Koščević, et al., 2019a) when comparing the angular error.

Exploring the effect of a scene's background in relation to a target object like Stanikunas et al., (2004), Flachot, et al., (2020) investigated the impact of different illuminants on classifying the reflectance of Munsell colour tiles and a Lambertian object in a set of digitally generated scenes and found that their three-layer CNN did not perform well when tested on different illuminants. Flachot, et al., (2020) highlight that whilst network performance only suffered slightly when removing the Munsell colour tiles from the scene, when the network was provided with a scene that had no background colour it failed completely. These findings are similar to those produced by Stanikunas et al., (2004).

Expanding network size and complexity, Afifi et al., (2021) trained a CNN on RAW images from a set of different cameras where the network used a set of images of completely different scenes from each camera to infer a spectral model for the input image. A hypernetwork, featuring 20 convolutional layers or varying sizes, is then used to create a colour constancy model for the input image that is then applied to correct for the illuminant (Afifi et al.,2021). Afifi et al., (2021) produced state-of-the-art results compared to multiple learning-based approaches across five different datasets (Cube+ Dataset, Cube+ Challenge, INTEL-TAU, Gehler-Shi, NUS) and producing angle errors as low as 1.77.

As per the above review, Illuminant estimation shows promise as a means of ensuring colour constancy across devices. This method of colour constancy does not, however, explicitly provide additional information about the target object within the scene, rather makes the object comparable across devices (Afifi et al., 2021) and over a range of illuminant spectra (Hu et al., 2017; Shi et al., 2016).

As such, the following section explores the estimation of multispectral and hyperspectral images in the form of reflectance data from their RGB representations as another method to address the illuminant and spectral sensitivity estimation aspect of the colour constancy problem.

2.6 Spectral Super-Resolution

Considering equation 2.9, colour is the basic term we use to describe the combination of these factors and whilst the colour of an object will change when the illuminant and/or camera sensor does so, its reflectance will remain the same as it is an intrinsic characteristic of the object itself (Kortum, 1969). Spectral reflectance data contains the maximum possible amount of information regarding how an object can and will appear under different lighting conditions and spectral sensitivities. That is, the set of all possible colours an object may appear can be estimated if the reflectance of an object is known by varying a scene's illumination (Flachot, et al., 2020) and the viewing camera's spectral sensitivity (Stiebel et al., 2018).

When high spatial resolutions are required, Reflectance data is typically captured using hyperspectral cameras (HSCs) (as opposed to a spectrophotometer) which respond to between ten to several hundred wavelengths of light, often both inside and outside the visible spectrum. Such spectra allow for the estimation of an object's internal properties such as firmness, sugar content, and many more characteristics of the object in a non-destructive manner (Schaare & Fraser, 2000). However, HSCs are expensive and often not suited to environments outside of the laboratory. As such, a field of research has arisen that explores the

estimation of reflectance from RGB images and is often referred to as single-image spectral super-resolution. This, in turn, means that when reconstructing spectra from RGB values, the outcome is a set of possible spectra as opposed to an individual spectrum. In other words, SSR is an indeterminate and highly unconstrained process (Alvarez-Gila et al., 2017).

Whilst HS imagery produces vast quantities of data, Arad & Ben-Shahar (2016) stipulate that these higher dimensional datasets can be represented sufficiently within a lower dimensional manifold. Other studies tend to confirm this supposition finding that higher dimensional datasets could be represented by between eight (Parkkinen et al., 1989) or between 13-23 (Hardeberg, 2002) components depending on size of the dataset and the spectra it represents. One solution to this indeterministic issue of SSR is the use of a prior (Arad & Ben-Shahar, 2016; Dong et al., 2016). A Prior refers to a sort of reference or dictionary that an algorithm can use to compare signals and spectra against to reproduce a more accurate hyperspectral spectrum.

The ill-posed nature of SSR also means it is prone to metamers. Arad & Ben-Shahar (2016) address the issue suggesting that the occurrence of metamers in lower dimensional (8 -23) manifolds is unlikely. Foster et al. (2006) provides a comprehensive analysis of the frequency of metamerism stating, "about 10^{-6} to 10^{-4} of all pairs of surfaces were indistinguishable under a day-light of correlated colour temperature 25,000 K and distinguishable by a certain criterion degree under one of 4000 K"

Galliani et al., (2017) implemented a CNN based on the architecture of Jegou et al., (2017) which features dense blocks developed by Huang et al., (2018). This network was shown to outperform (Arad & Ben-Shahar, 2016) as well as a radial bias-based approach by (Nguyen et al., 2014), producing a root mean square error (RMSE) of 1.980, 4.76, and 5.27 on the Imperial Computer Vision Lab (ICVL), Columbia Computer Vision Lab (CAVE), and NUS datasets respectively.

In comparison, Aeschbacher et al., (2017) argue that a CNN approach is not necessary in SSR, opting instead to create an overcomplete sparse dictionary using k-means clustering and singular value decomposition (K-SVD). This method for SSR, named A+, showed better results than Galliani et al., (2017) on both the ICVL and NUS datasets, but poorer performance on the CAVE dataset.

Weijer & Garrote, (2018) applied a generative-adversarial network (GAN) to the problem of SSR and report an RMSE or 1.457 compared to 1.980 (Galliani et al., 2017) and 2.633 (Arad & Ben-Shahar, 2016) on the ICVL dataset. Can & Timofte, (2018) produced a similar improvement on the works of Galliani et al., (2017), Arad & Ben-Shahar, (2016), and Aeschbacher et al., (2017) by incorporating a series of residual blocks into their CNN. This trend of deeper and larger CNNs and accompanying improvement in results continues in the works of Yan et al., (2018), Xiong, et al., (2017), and Shi, et al., (2018).

The state-of-the-art regarding single-image spectral super-resolution is best derived from the results of the NTIRE 2020 competition (Arad et al., 2020). The competition featured two datasets, one of clean data created from noise-free RGB images created from hyperspectral imagery and known spectral sensitivities, and a 'real world' dataset featuring images from an unknown device with JPEG compressed RGB images (Arad et al., 2020).

In the competition, Li et al. (2020) ranked first in the 'clean' dataset and third in the 'real world' dataset using an adaptive weighted attention network (AWAN) which utilises a prior constructed from different camera spectral sensitivities. Similarly, Zhao et al., (2020) placed first in the 'real world' and third in the 'clean' datasets by using a hierarchical regression network (HRNet). These networks scored RMSE values of 0.01293 and 0.01389 for the 'clean' dataset and RMSE values of 0.01923 and 0.01991 for the 'real world' dataset

respectively. Keep in mind the previous RMSE results in the above review have been on different datasets hence the change in magnitude of the error for the NTIRE competition.

Several methods have been applied to achieve single-image SSR although modern methods almost exclusively use CNNs (Stiebel & Merhof, 2020). Stiebel & Merhof (2020) note that even the state-of-the-art methods for SSR show a susceptibility to changes in brightness and go on to propose a method to train CNNs using a signal's direction vector as opposed to a signal's absolute magnitude as is normal practise. Whilst their system does not provide a better result than other CNNs, it does show a robustness to reproducing spectra under varying lighting conditions (Stiebel & Merhof, 2020).

In terms of the application of SSR, Park et al. (2020) demonstrated the successful use of SSR in predicting patients' blood haemoglobin levels and found that their model successfully predicted haemoglobin levels with an accuracy of 95%. Similarly, Sharma & Hefeeda (2020) used the HSCNN+ network proposed by Shi et al., (2018) to reconstruct SSR images of human veins for use in point-of-care applications showing promising results.

Zhao et al. (2020) demonstrated the use of SSR in estimating the reflectance of tomato using the HSCNN-R network developed by Zhao et al. (2020) and compared the reflectance estimates from images taken from a smartphone camera (Samsung S9) against reflectance's computationally generated from hyperspectral images. Zhao et al. (2020) found that even when the smartphones spectral sensitivities were not directly incorporated into the training dataset, the estimated spectra had an RMSE of 6.223 over a whole image.

The above review highlights that SSR can be achieved with low errors. In particular, Zhao et al. (2020) show the SSR can be readily applied to estimate the reflectance of tomatoes. There does not appear to be any reason the that SSR is not suitable for application in estimating the reflectance of red kiwifruit.

2.7 Developing a Colour Measurement Scale

Considering this project will explore creating a new scale of redness this section provides an exploration of previous work in this area. It is common to use ΔE to compare colours in colourimetry research as it provides a single value to represent colour accuracy. However, ΔE requires two colours, a ground truth, and an estimate/prediction, to be calculated. This ground truth value is commonly measured using colourimetric equipment such as a spectrophotometer.

Fruit maturity is often correlated to colour. Fruit maturity scales, especially those related to colour, are typically created by industry experts, and then utilised by researchers to create regression models for computer vision applications (Zhou, et al., 2021; Avilaa, et al., 2015; Hasan & Monir, 2017). When developing a maturity scale, fruit are typically imaged over their entire growing cycle and the relative changes in colour are analysed and correlated to maturity (Avilaa, et al., 2015). In some cases, fruit are classified into different maturities based on the age and size of the fruit (Prabha & Kumar, 2015) and in other cases, fruit are labelled subjectively by the researcher(s) (Kaur, et al., 2018) before colourimetric analysis.

Zespri has provided this project with their current Red19 colour assessment guidelines although the inception of this project was predicated on the fact that the current guidelines produced noisy and unreliable datasets. As such, this project will explore relating kiwifruit redness to consumer preferences as opposed to just experts in the hope that this will lead to better outcomes on the market-side of the supply chain.

2.8 Summary of Findings

Whilst colour constancy is a promising field or research for enabling the objectification of kiwifruit colour quality, there are some potential shortcomings to using traditional colour constancy methods. To the best of our knowledge kiwifruit colour has never been considered in in both spatial and spectral dimensions, and therefore it is not yet known what level of sensitivity is needed to accurately differentiate kiwifruit regions. That is, how large a change in reflectance will lead to a change in a kiwifruit's colour grade both perceptually and in relation on the current grading scale.

Considering the objective of this project: *to explore, identify, and assess a method and scale to measure the colour quality of Red19 kiwifruit that can be deployed and utilised on any smartphone under any lighting conditions*, some conclusions can be drawn on the methods of colour constancy reviewed in the previous sections. This project may lead to the development of a deployable system used by the kiwifruit industry and as such, consideration is given to whether each solution is able to be scaled in a sensible manner.

Table 2.1: Summary and critique of literature review findings

Method	Suitability	Consideration
Subjective Observer (current system)	Low	Currently produces a noisy and inconsistent dataset that makes quality control difficult.
Spectrophotometer/Colorimeter	Low	Current systems do not have sufficiently high spatial resolution for Red19 kiwifruit.
Controlled optical enclosure	Low	Not scalab–e – would require the production or purchase of thousands of enclosures, does not address differing spectral sensitivities.
Standard Illuminant	Low	Not scalab–e – would require the purchase of thousands of standard illuminants, does not address differing spectral sensitivities.
Smartphone Attachment	Low	Not scalab–e – would require the design and production of thousands of bespoke enclosures, does not address differing spectral sensitivities.
Standard colour space (sRGB)	Low	Produces a noticeable perceptual difference when using different smartphones and different lighting conditions.
Generic Colour Calibration Target	Moderate	Produces a noticeable perceptual difference using only one camera and a standard illuminant (Wang & Zhang, 2013) Not scalab–e – would require the purchase and of thousands of CCTs.
Custom Red Kiwifruit Colour Calibration Target	Moderate	Produces a noticeable perceptual difference using only one camera and a standard illuminant (Wang & Zhang, 2013) Not scalab–e – would require the design and production of thousands of bespoke CCTs.
Classic White Balancing Techniques - Gray Worlds - White Patch - Gamut Mapping	Moderate	Produce reasonable results in most situations although often produce a noticeable perceptual difference from the ground truth value/colour.
CNN-based Illuminant Estimation	High	Produces excellent results across a range of smartphones and illuminants.
CNN-based Spectral Super Resolution	High	Produces excellent results across a range of smartphones and illuminants

There is an argument for applying either illuminant estimation or SSR in the case of Red19. Whilst illuminant correction would allow for comparable results across a range of illuminances, a highly complex solution like

that of Afifi et al., (2021) would be needed to sufficiently account for the spectral sensitivities of different smartphones. Similarly, current state-of-the-art in SSR applies CNNs in an increasingly complex way to achieve SSR across a set of general scenes. This begs the question; will better colour constancy be achieved through illuminant and sensitivity estimation or reflectance estimation alone? As these solutions solve the colour constancy problem through explicit estimation of different variables in equation 2.9.

It should be noted that previous SSR methods have shown a susceptibility to changes in brightness although Stiebel & Merhof, (2020) have shown a promising solution to this issue. More so, satisfactory SSR could not be achieved without implicitly learning invariance to spectral sensitivity and illumination in some form. SSR does provide an enticing solution to this problem in two ways: 1, it seeks to estimate a characteristic that is intrinsic to the fruit itself rather than the conditions that is in photographed in; and 2, because reflectance data contains information regarding other internal characteristics/properties of the fruit itself, this method could be extended beyond just colour quality in future research.

Previous studies of SSR have used deep and large convolutional networks to achieve SSR in general and widely varying scenes. In the case of Red19, a much smaller and lighter model is likely sufficient as it needs only to learn the estimation of reflectance regarding red kiwifruit – not a range of fruits or scenes. As such, it is suggested that the most promising solution may be by way of a small and interpretable CNN that can estimate the reflectance of a kiwifruit or a set of kiwifruit regions and a classification system to match this reflectance data to a final kiwifruit grade. This solution also allows for understanding of how the system is reaching its final score or grade as the estimation of reflectance is an explicit part of the system.

In the case of Red19 kiwifruit, there are three different regions of colour to consider (Outer pericarp, inner pericarp, and core), the gradients between them, and the texture of the fruit that affects the presentation of redness and the customer's perception of the fruit. Similarly, whilst many researchers are constrained by a lack of data this project has a set of 1800 hyperspectral images each paired with four RGB images (7200) with the RGB images taken by two different smartphones (Samsung S8 and iPhone 12). As such, the constraints of previous studies relating to a lack of data should not affect this project. Similarly, it may be the case that some of the previous studies' methods that were constrained by a lack of data may perform better given the depth of this project's dataset.

Therefore, this study will explore the development of a colour assessment models using spectral super-resolution. Initially, the study will explore the estimation of the average reflectance for a kiwifruit from RAW images as well as its three key regions (core, inner, and outer pericarp). Finally, the study will explore how to extract a redness score from this spectral data and how this might be implemented on a typical smartphone.

3.0 Methodology

Kiwifruit spectra is best captured using a hyperspectral camera. To calibrate a model to estimate the reflectance of Red19 kiwifruit, our methodology has consisted of capturing a series of RGB/RAW images paired with their ground truth reflectance data from hyperspectral images.

The key methods to achieve the goal of this project are as follows:

1. Capture hyperspectral and RGB/RAW images.
2. Develop a new metric/scale to represent the redness of Red19 kiwifruit.
3. Train a model to estimate the reflectance of Red19 Kiwifruit.
4. Train a model to predict the redness score of Red19 Kiwifruit from its reflectance data

3.1 Hyperspectral Imaging

3.1.1 Fruit Collection

1863 fruit was collected from 11 different growers from across New Zealand. Fruit was collected at harvest maturity and transported to Massey University where it was stored at 1°C in a controlled temperature room (CTR) until it was imaged. Kiwifruit was stored in plastic poly-liner packaging placed within cardboard boxes as provided by Zespri. Prior to image capture and due to the larger CTR that held all of the fruit at 1°C being in different building from the HSC, small portions of kiwifruit were transported to a CTR nearer the HSC system which was held at 3°C. Kiwifruit were not stored more than three days in the CTR near the HSI system. Data collection occurred over a period of 27 days between the 1st of April and April 27th, 2021.

Of the 11 growers, four growers have fruit that span the majority of the different fruit sizes (27844, 5205a, 48785, 7829e). The other 7 growers have fruit only of size 36 (Figure 3.1).

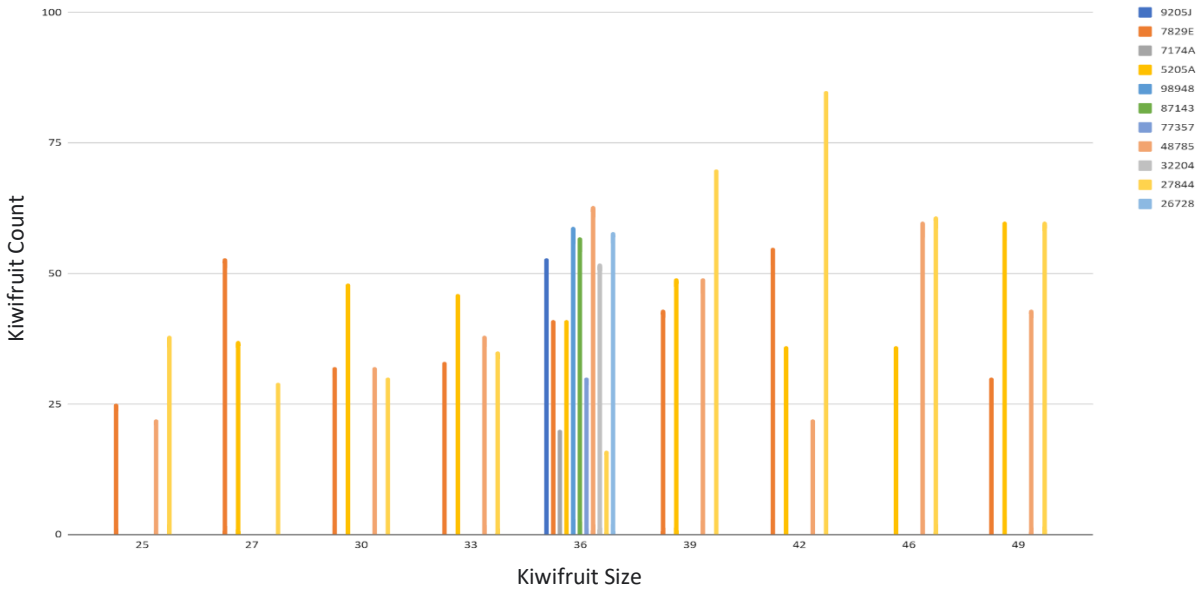


Figure 3.1: Distribution of Kiwifruit count by grower and fruit size.2 Image Capture

3.2.1 Image Capture

HS images were captured using a Specim FX10 push-broom camera (Specim Spectral Imaging Ltd, 2021) translation stage, and lighting rig featuring two arrays of seven halogen bulbs oriented 45° to the translation stage. The FX10 is an all-in-one imaging solution designed for industrial and laboratory use with a spectral range of 400nm-1000nm and a sampling distance of 2.7nm producing 224 bands over its spectral range. The FX10 has a resolution of 1024 pixels per line. The HS imaging rig (Figure 3.2) was setup in a dark room.

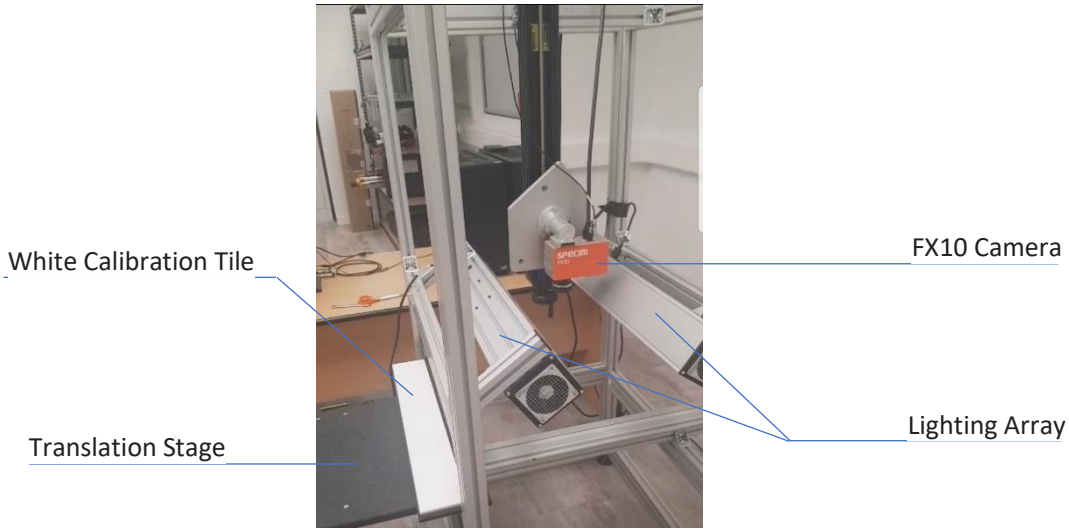


Figure 3.2: Hyperspectral Imaging Rig.

To maintain consistency across fruit sizes, each fruit was cut into a 30mm slice using a custom-made, 3D-printed guillotine mechanism (Figure 3.3). A 30mm slice was taken so the depth of the slice exceeded the light penetration depth of kiwifruit as estimated by Qin & Lu (2008).

The height of the camera was set to 390mm above the translation stage as this was the lowest the rig would allow.

A custom-made, laser-cut iris mechanism was created which straddled the translation stage and ensured the center of each fruit aligns approximately with the center of the imaging lens (Figure 3.4)

Spatial calibration is undertaken by running the automatic calibration algorithm (Middleton Spectral Vision, 2021). This algorithm adjusts either the frame rate of the camera or conveyor speed of the translation stage to ensure each image maintains the correct aspect ratio. Four conveyor speeds were tested by running the calibration algorithm until it reached a steady state (produced the same suggested settings each time).

Figure 3.3: 3D-printed Guillotine Mechanism used to slice cross sections of Kiwifruit.

Upon visual inspection, a lower conveyor speed and lower frame rate appear to produce more realistic results. That is, results that appear to maintain the correct shape as seen by the human eye. Higher speeds appear to stretch the images slightly. The translation speed was set to 5mm/s and the camera frame rate was set to 20.72Hz.

Exposure time was determined experimentally by imaging the same fruit and varying exposure time. An exposure time above 5.5ms showed clear saturation on the white calibration tile of ~99% reflectance and hence this was taken as above the maximum limit for exposure time. Because exposure varies as a function of wavelength, longer exposure times are favourable in HSI as it allows for more information to accumulate on the sensor and thus improves the quality of the data (so long as it is not saturated).

Figure 3.4: Iris Mechanism used to Kiwifruit during hyperspectral image capture.

Figure 3.5 shows that at 5.5ms there is a clear saturation of the maximum value around the 120th band. Similarly, 5.2ms also reaches this saturated value. 5ms exposure time appears to offer maximal pixel intensity without reaching saturation.

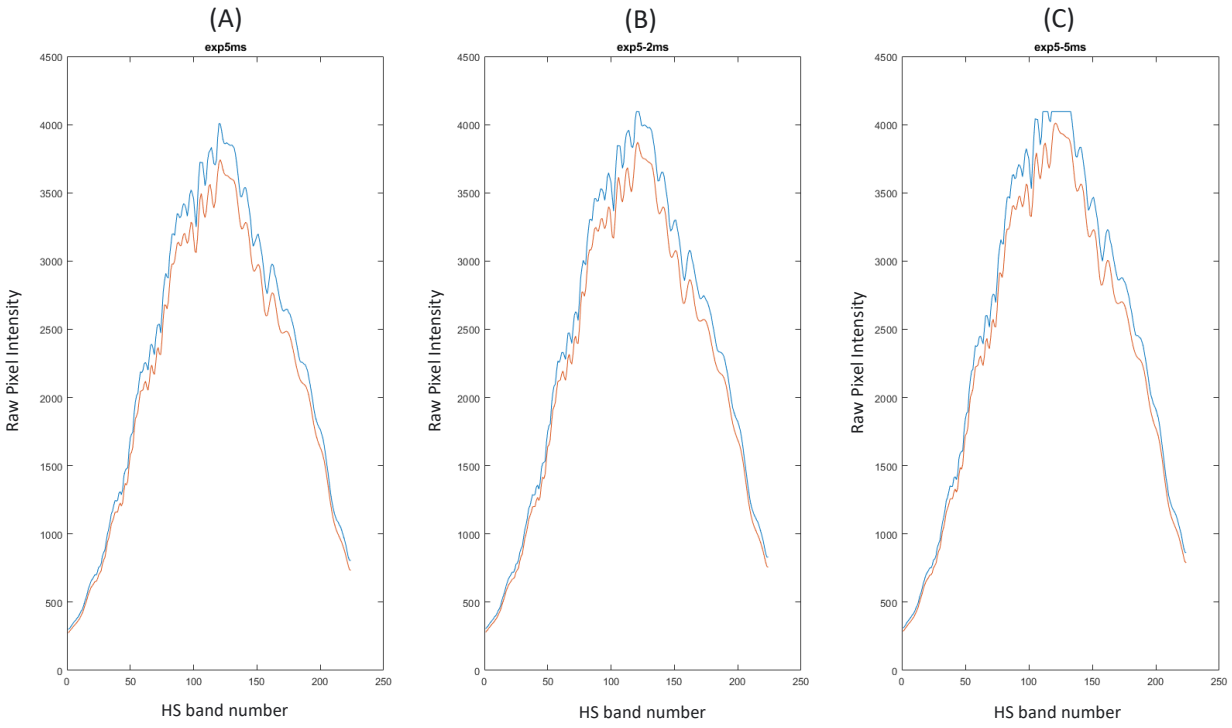


Figure 3.5: Comparison of mean (orange) and max (blue) signal from FX10 by band number for exposure times of 5 ms (A), 5.2 ms (B), and 5.5 ms (C) where lower band numbers correspond to shorter wavelengths starting at 400nm – 100nm.

Before images were captured the HS camera was given at least 30 minutes to warm up to allow the sensor and light bulbs to reach a steady state temperature. A set of 10 images were captured to produce an estimate of the signal to noise (SNR) for the HSI system and was found to have an average SNR of ~115 and a median of SNR ~90 (Figure 3.6).

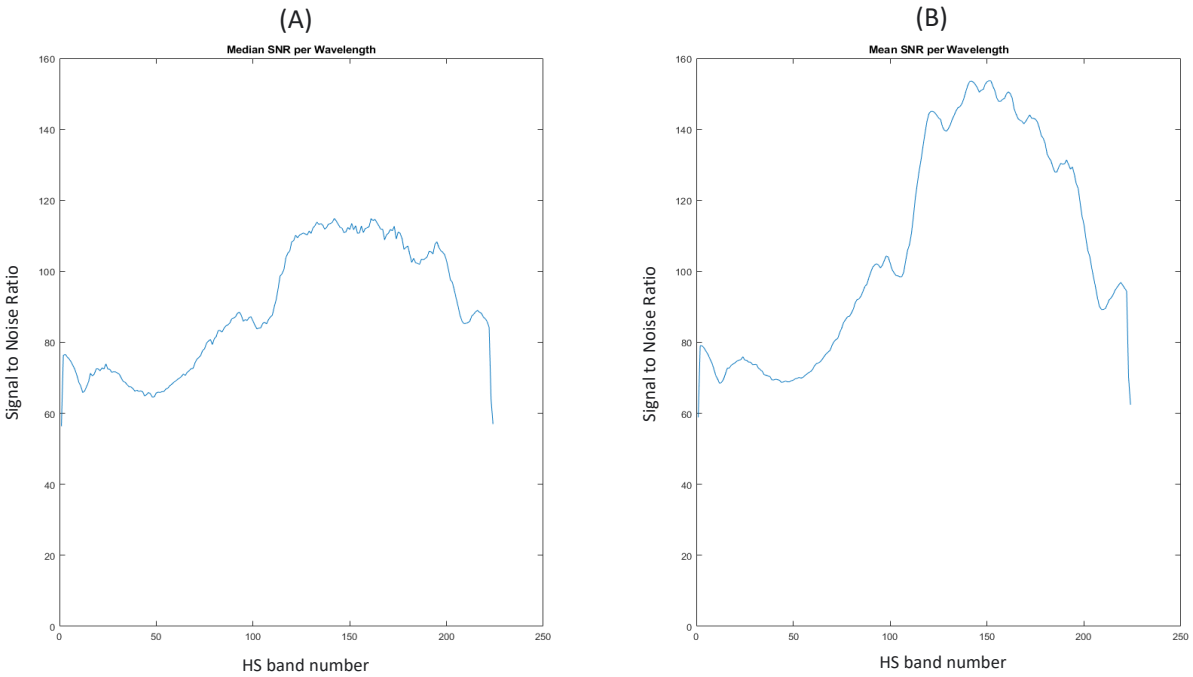


Figure 3.6: Comparison of median (A) and mean (B) signal to noise ratio for a single kiwifruit image.

3.1.3 Image Processing

Each HS image capture produces three RAW image files: one of the white reflectance tile, one of the dark current (shutter closed), and one of the kiwifruit. Each RAW image has a header and log file associated with it. The imaging software also outputs a series of metadata and calibration files for each image. A general overview of the image processing pipeline is shown in figure 3.7.

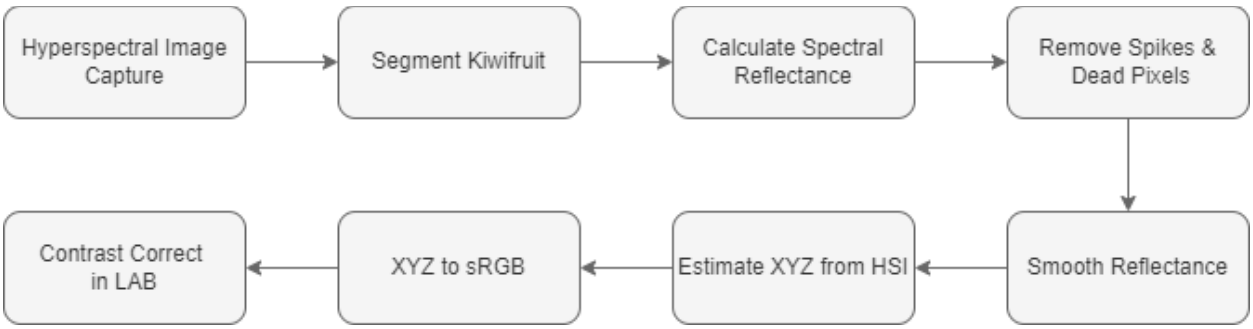


Figure 3.7: Overview of image processing processes to convert raw hyperspectral images into contrast-corrected sRGB images.

Due to the fixed height of the camera rig, approximately 250 of the total 1024 pixels contained information related to each kiwifruit. As such, a segmented script was written to automatically extract the data only related to each kiwifruit from each HS image. This script used the false colour images from the HS data and segmented the fruit based on the intensity of the blue colour channel to produce a binary mask that could then be used for segmentation purposes (Figure 3.8)

(A) (B)

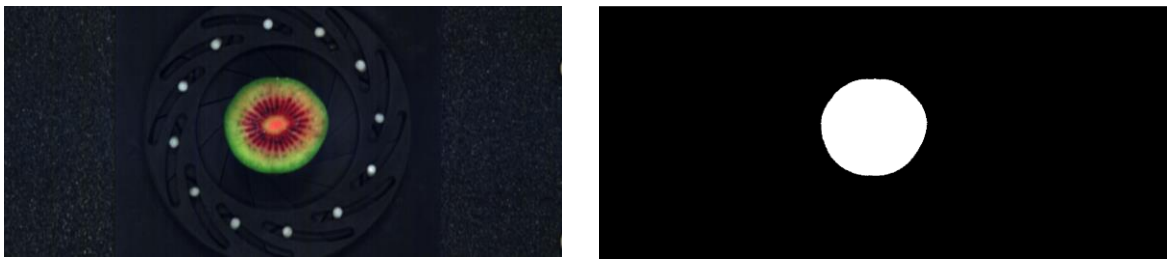


Figure 3.8: False Colour Image produced by hyperspectral camera (A) and Segmentation Mask (B) for a single kiwifruit.

Spectral Reflectance

Spectral reflectance is the ratio of incident light that is reflected by an object's surface (Kim, Chen, & Mehl, 2001). Therefore, the white tile and dark current image can be used to convert the intensity information from the kiwifruit image into its reflectance data using the following equation.

$$R_{i,\lambda} = \frac{I_{i,\lambda} - Rd_{i,\lambda}}{Rw_{i,\lambda} - Rd_{i,\lambda}}$$

Equation 3.1: Calculating Reflectance from pixel intensity values

Where $R_{i,\lambda}$ is the reflectance value, of between 0 and 1 for pixel i for a given wavelength λ. Where Rd is the dark current intensity, Rw is the white tile intensity, and $I_{i,\lambda}$ is the kiwifruit image intensity (Kim, Chen, & Mehl, 2001).

Spike & Dead Pixel Removal

Spike and dead pixel removal is important to mitigate against unwanted effects from instrument failure. It is estimated that up to 1% of pixels in an image may suffer from spikes or dead pixels (Dorrepaal, Malegori, & Gowen, 2016).

Dead pixel/spikes were detected using a custom piece of code written in Matlab. In an HS image, neighbouring pixels' spectra should be similar as spectra observed in the natural world tend to be smooth (Dorrepaal, Malegori, & Gowen, 2016).

To detect spikes, each pixel's spectra was subtracted from its neighbours' spectra and the resulting difference graph analysed. A threshold for what constitutes a spike was set through trial and error and a nine-point moving average filter was implemented to interpolate information from either side of a spike and correct the error (Figure 3.9)

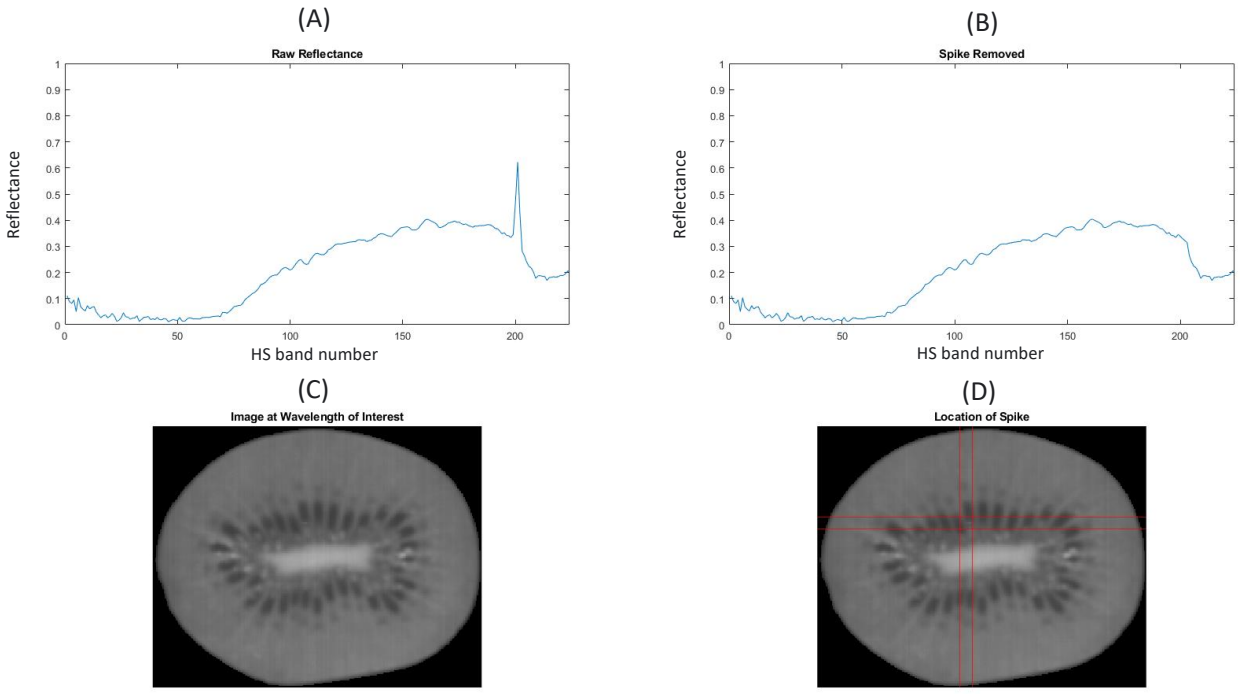


Figure 3.9 Spectra (A) and kiwifruit image (C) before spike removal vs spectra (B) and kiwifruit image (D) after spike removal.

Scatter Correction & Smoothing

Scattering is the term used to describe the mixing of light due to the reflection, refraction, and diffraction by an object's surface (Kortum, 1969). There are two predominant types of scattering; single and multiple scattering. To correct for this effect researchers typically use multiplicative scatter correction (MSC), or standard normal variate (SNV) correction.

MSC involves fitting a first order polynomial between the spectra of interest and the average spectra of the set of all samples.

For MSC, where $X_{i,j}^{MSC}$ is the corrected point j of the i^{th}spectra of the sample

$$X_{i,j}^{MSC} = \frac{X_{i,j} - a}{b}$$

Where $X_{i,j} = a + b(y_{avg}) + E$

And y_{avg} is the average spectra across all samples

Equation 3.2: Calculating multiplicative scatter correction

For SNV, where $X_{i,j}^{SNV}$ is the corrected point j of the i^{th}spectra of the sample

$$X_{i,j}^{SNV} = \frac{X_{i,j} - \overline{X_i}}{Sd}$$

Where $\overline{X_i}$ is the mean and Sd is the standard deviation of the i^{th} spectra.

Equation 3.3: Calculating standard normal variate correction

Another method that addresses both scattering effects and the smoothing of noisy data is the Savitzky-Golay moving average filter (Manley, 2014). This was applied using MATLAB's *smoothdata* function (The MathWorks Inc, 2021). The optimal filter size was determined empirically.

To the best of our knowledge, there does not appear to be a standard practise for filtering spectra when trying to preserve colourimetric fidelity. Therefore, we are left in a dichotomy whereby spectra observed in the natural world tend to be smooth (Dorrepaal et al.,2016), but the most accurate colourimetric representation of our kiwifruit is generated from noisy, raw reflectance spectra (Figure 3.11).

Attempting to filter the spectra without creating a residual proved fruitless as minimal filtering produced a residual. To compromise between spectral smoothing and colourimetric fidelity, the average ΔE between the raw and filtered image was measured across a range of filter sizes. A suitable filter size was determined to be 15 which produced an average ΔE of 1.3 across the whole kiwifruit whilst removing most of the high frequency noise in the spectra (Figure 3.10 & 3.11).

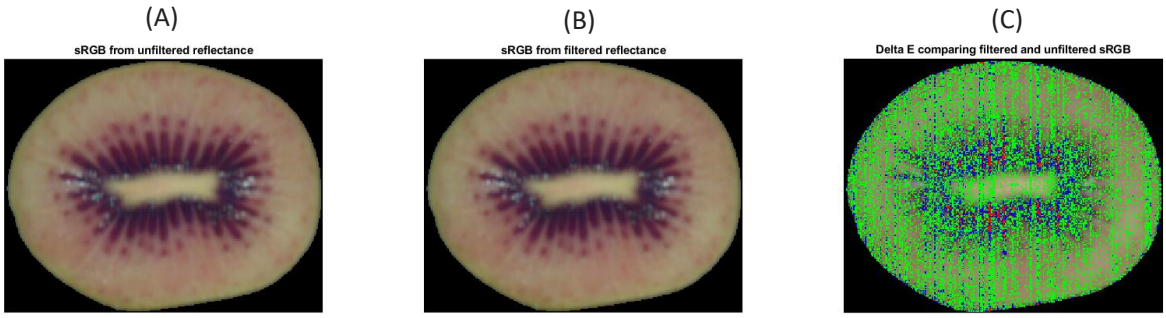


Figure 3.10: Comparison of sRGB kiwifruit before (A) and after (B) spectral smoothing with highlighted differences in ΔE (C) where green (<1), blue (<2.3), red (>4).

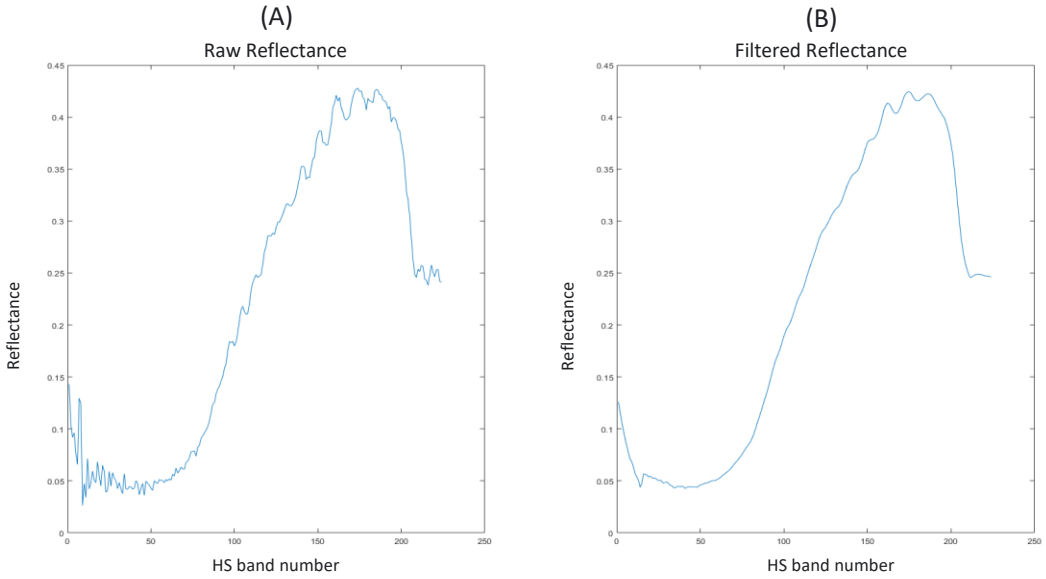


Figure 3.11: Comparison of raw (A) vs smoothed (B) kiwifruit spectra

Whilst SNV and MSC have been widely applied in HSI, there is little information on how these techniques affect the colour fidelity. To address this lack of information, a small experiment was undertaken to compare the results of each technique.

SNV and MSC were processed from the filtered HSI. However, because SNV shifts the mean of the spectra to zero, both outputs from the correction techniques were adjusted so that their means matched that of the filtered data (Figure 3.12). sRGB images were then created from each set of spectra and contrast adjusted (Figure 3.13). These results show that SNV and MSC are not suited for this application as they comprise the colour information of the kiwifruit images and look nothing like the real kiwifruit.

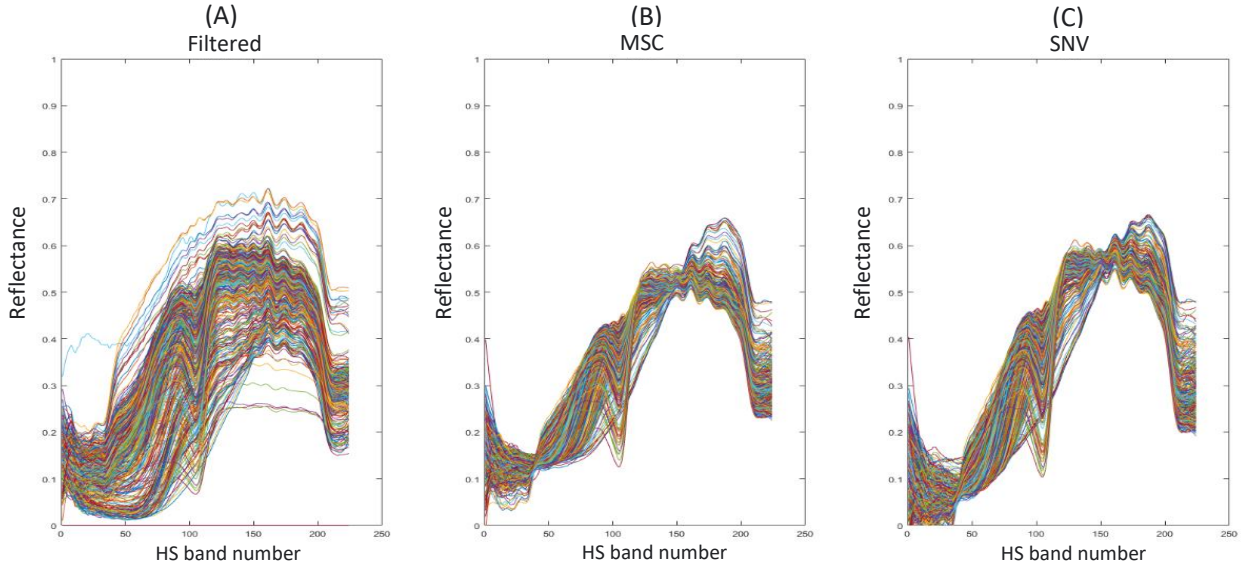


Figure 3.12: Comparison of spectra when filtering (A), using MSC (B), and SVN (C) correction techniques.

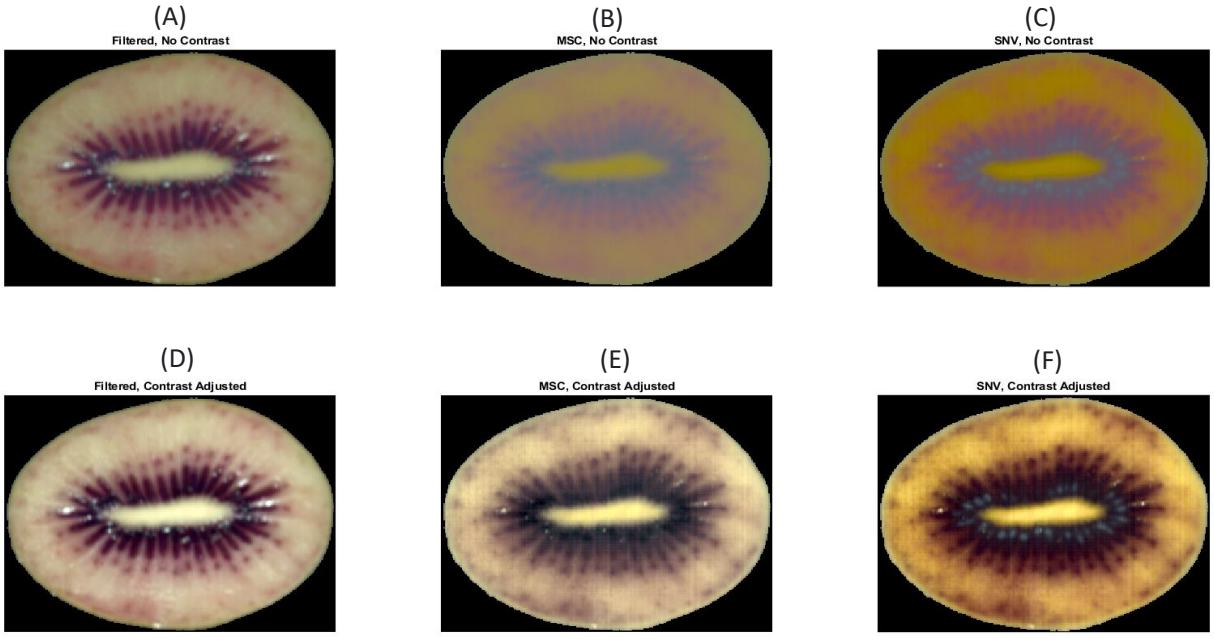


Figure 3.13: Comparison of kiwifruit images after spectral correct techniques and before contrast correction: filtered (A), MSC (B), and SNV (C) and kiwifruit images after contrast correction: filtered (D), MSC (E), and SNV (F).

3.2 RGB/RAW Image Capture

RAW images were captured using the Adobe Lightroom CC (Adobe, 2021) mobile application on both a Samsung Galaxy S8 (Samsung, 2017) and an iPhone 12 (Apple Inc, 2020) mobile phone. A custom bracket was designed and then 3D-printed which held each phone perpendicular to the HS system's translation stage at a height of ~50cm (Figure 3.14).

Figure 3.14: 3D-printed smartphone bracket over hyperspectral imaging system.

The image capture settings are shown in table 3.1. The capture settings differ slightly as Adobe Lightroom CC takes input for these settings using a slider. Whilst convenient, the slider is incredibly difficult to set exact numbers with hence why shutter speed differs slightly. Likewise, the ISO was set to the nearest selectable option in the android and IOS app.

Table 3.1: Camera capture settings

Camera Setting	Samsung Galaxy S8	iPhone 12
Shutter Speed	1/930	1/1000
ISO	50	40
White Balance	Incandescent	Incandescent
Focus	Auto	Auto

Each fruit was imaged from the phone bracket as well as an angle of approx. 45° by removing each phone from the bracket and holding it in-line with the structure of the test rig.

3.3 Sensory trial

To produce a meaningful measure of redness, a small sensory trial was undertaken to link a quantifiable measure of redness to actual consumer preferences. This was done so that the final 'judge' of redness is the consumer with the supposition that tying consumer preferences to a scale of redness will likely produce better outcomes than compared to only an expert opinion alone.

3.3.1 Reflectance to AdobeRGB & sRGB

With the spectral reflectance data of each kiwifruit, a series of scenes can be created of what each kiwifruit would look like under different illuminants and spectral sensitivities. To create meaningfully comparable images, a series of images were created using the CIE standard colourmetric observer (1931) and the D65 illuminant.

Images were created following (Magnusson, et al., 2020) by estimating CIE XYZ values from reflectance data and then transforming these values into sRGB or AdobeRGB. As per equation 3.4, a transformation from XYZ to sRGB or AdobeRGB can be completed using a transformation matrix T such that,

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = T \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Equation 3.4: XYZ to sRGB

Where M is a 3x3 colour transformation matrix which is

$$M_{sRGB} = \begin{bmatrix} 3.2406255 & -1.5372080 & -0.4986286 \\ -0.9689307 & 1.8757561 & 0.0415175 \\ 0.0557101 & -0.2040211 & 1.0569959 \end{bmatrix}$$

(Magnusson, et al., 2020)

and

$$M_{AdobeRGB} = \begin{bmatrix} 2.04159 & -0.56501 & -0.34473 \\ -0.96924 & 1.87597 & 0.04156 \\ 0.01344 & -0.11836 & 1.01517 \end{bmatrix}$$

(Adobe Systems Incorporated, 2005)

Each image was then lightness-contrast adjusted by first converting each image into the CIE LAB colour space (D65 illuminant/standard observer) and then using MATLAB's *imadjust* function to adjust the L channel based its histogram distribution.

3.3.2 Experimental Protocol

A visual sensory study was designed to corelate consumer preferences of redness to a measurable value. RGB images were created in Adobe1998 colour space from reflectance data and then shown to participants. Data collection began on the 7th of Oct and finished on the 21st of October. A total of 34 participants were recruited.

Recruitment

Due to ongoing Covid-19 restrictions, Individuals were recruited from within the University and its immediate surroundings to limit contact between individuals who are not usually in close proximity. Individuals had to be over the age of 18, not *dislike* kiwifruit, and not be a 'high-risk' individual (have underlying health conditions or residing with someone who does) as per Massey University's Covid-19 Guidelines to be eligible for the study. Each participant was prompted to undertake an Ishihara test (Ishihara, 1972) at the beginning of the experiment to ensure they had normal colour vision.

Participation

The sensory trial was completed using the QuickEval software (Norwegian Colour and Visual Computing Laboratory, 2021) created by the Norwegian University of Science and Technology. The experiment was designed as a balanced incomplete block design where each block consisted of four treatments and each treatment consisted of a set of 12 kiwifruit images (Table 3.2). A balanced incomplete block design was chosen as it allows for the inclusion more treatments (fruit sets) with a moderate number of participants.

The experiment was conducted in a completely dark room using a colour-calibrated monitor (EIZO Corportation, 2016) that was calibrated to AdobeRGB colour space and a colour temperature of 6500K.

Kiwifruit images in AdobeRGB and 6500K were then shown as a paired comparison experiment and the participants was asked to '*select the kiwifruit that appears most red'* to them (Figure. 3.15)

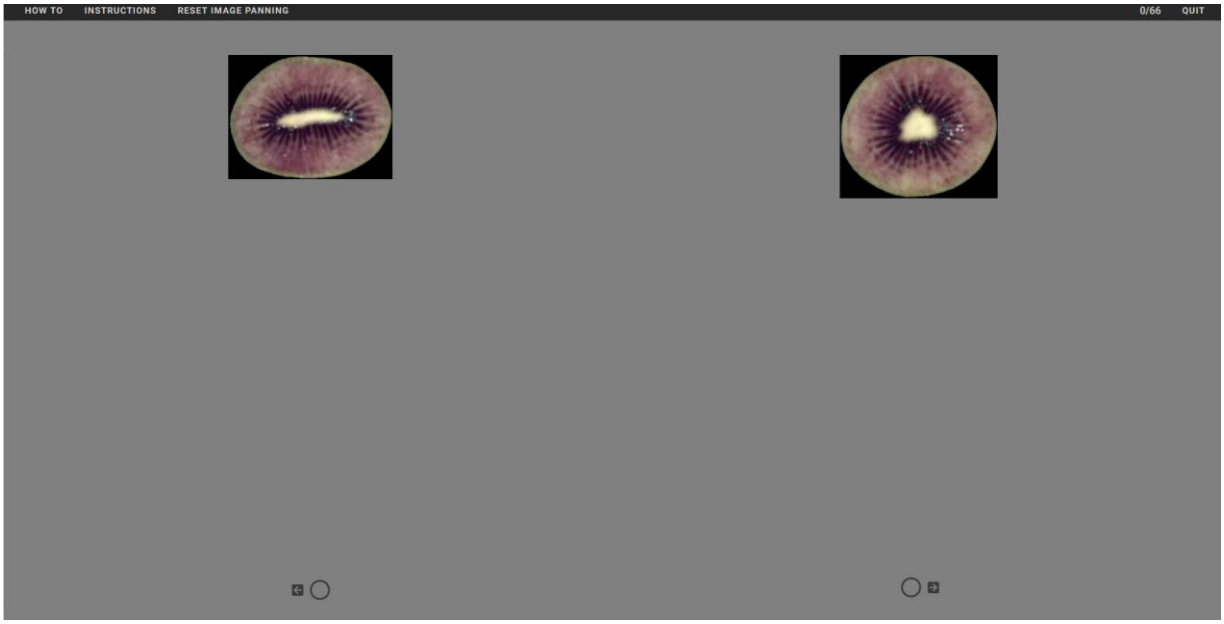


Figure 3.15: Example of what participants saw when undertaking a single paired comparison using QuickEval website.

Pseudo Redness Scale

In a paired comparison experiment, the most meaningful comparisons are those that are closely related to each other in relation to the characteristic in question (Tsukida & Gupta, 2011). That is, comparing a very red and very green kiwifruit is meaningless as the outcome is already known. Rather, comparing a moderately red kiwifruit with a slightly less red kiwifruit allows the exploration of the extent of colour difference needed before one kiwifruit is considered redder than another and at which point people will (almost) always agree one fruit is redder than another.

A rough redness scale was created using the AdobeRGB images by taking the average of the red RGB channel and arranging the fruit along this scale. Visually, this did not provide a smooth scale of redness. Further exploration revealed that fruit which appear most red appear this way not due to the absolute intensity of their red pixels, but rather their lack of intensity with respect to their green channel (Figure 3.16). Therefore, redness of Red19 kiwifruit is best considered as a lack of green as opposed to a higher redness. This finding is supported by the discussion in section 2.1.1 where figure 1.4 shows that hue is derived from the difference in the excitation of the medium (green) and long (red) cone cells in the eye (Hunt & Carvalho, 2016; Lee & Silveira, 2016). We propose that it is reasonable that this excitation difference in medium and long cone cells

may be roughly equated the perceptual redness of kiwifruit.

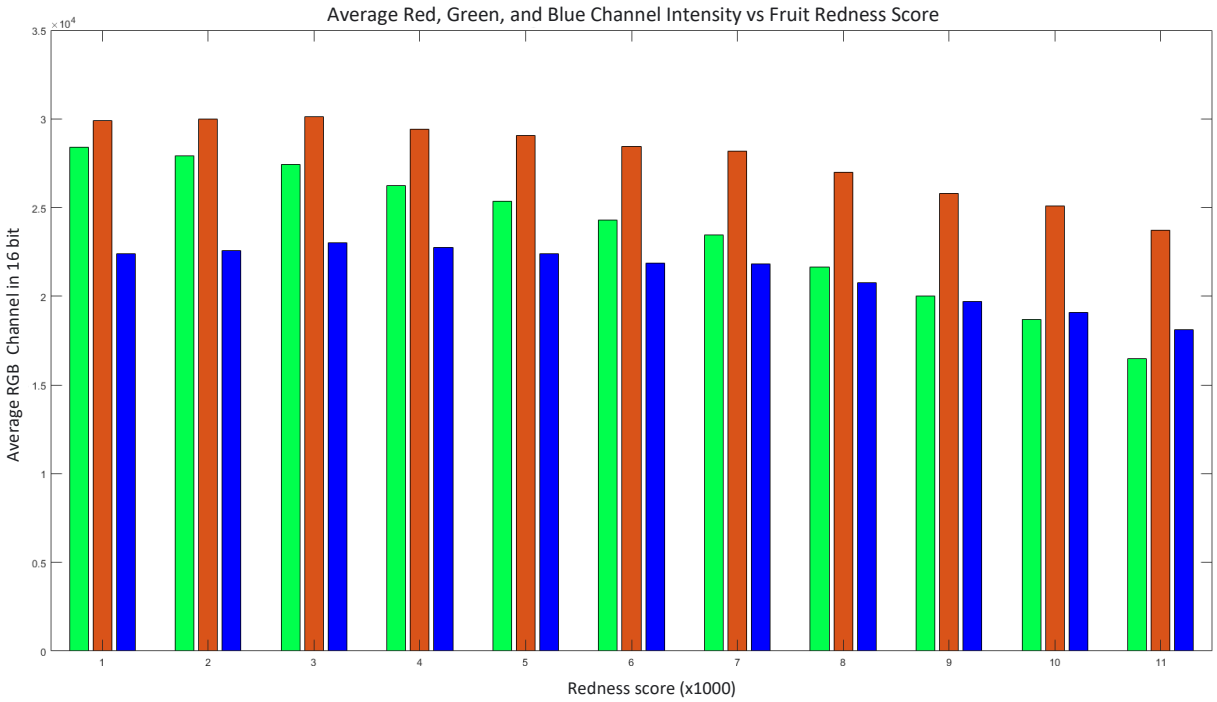


Figure 3.16: Comparison of average red, green, and blue channels of kiwifruit by redness score. X axis represents the redness score of kiwifruit in 1000 step increments where low scoring fruit appear green and higher scoring fruit appear red. The Y axis red, green, and blue bars represent the average intensity of the red, green, and blue channel of kiwifruit images. The figure shows that as perceptual kiwifruit 'redness' increases (score increases), the average intensity of the red and green sRGB channels decreases

Considering this result, a new scale was created which considered the distance of the average red channel from the average green channel of each kiwifruit. This scale represents an ad hoc representation of overall kiwifruit 'redness'. This produced a visually smooth scale from very green fruit to very red fruit over a range of -2000 (very green) to ~12,000 (very red) from 16bit images (Figure 3.17).

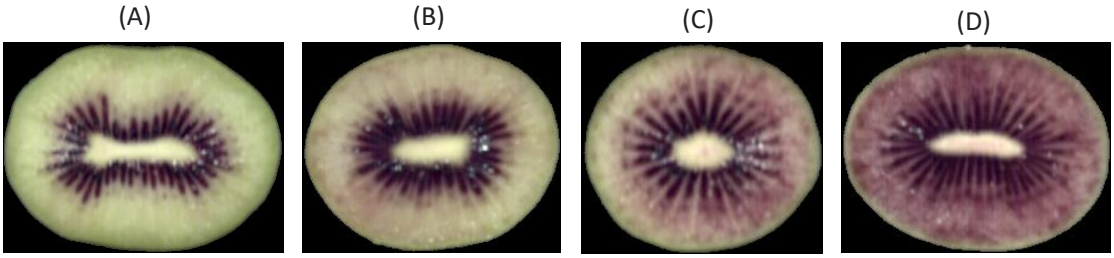


Figure 3.17: Comparison of Kiwifruit from very green to very red arranged by redness score from 0 (A), 3000 (B), 6000 (C), 9000 (D)

Using this new scale, sets of kiwifruit were created based on their redness score whereby the distance or difference in redness score was used to explore how similar and dissimilar redness scores need to be before participants agree that one fruit is redder than another. Table 3.2 shows the design of this experiment. Each block of experiments consists of four treatments i.e., block 5 consists of treatments A, B, C, and D, where each treatment (A, B, C, D) are a set of 12 kiwifruit. Table 3.2 shows the order each participant completed each treatment within their respective block. Each block was undertaken by no less than five participants.

Table 3.2: Balanced Incomplete block design for Kiwifruit sensory trial. Each block is made up of four treatments (A, B, C, D) where each treatment is a pair comparison experiment of 12 kiwifruit images. Each block was undertaken by 5 participants and the treatment order was randomised.

Participant ID	Block Number	Set 1	Set 2	Set 3	Set 4
23	5	C	D	B	A
24	5	C	B	D	A
25	5	A	C	D	B
26	5	D	B	C	A
27	5	D	C	A	B
28	6	E	F	G	H
29	6	H	G	F	E
30	6	H	F	E	G
31	6	H	E	F	G
32	6	E	F	G	H
33	4	I	J	K	L
34	4	J	I	K	L
35	4	K	L	I	J
36	4	L	K	J	I
37	4	I	J	K	L
38	7	N	P	M	O
39	7	O	P	M	N
40	7	P	M	O	N
41	7	M	P	N	O
42	7	O			
43	7	O	N	P	M
44	8	Q	S	T	R
45	8	S	Q	T	R
46	8	T	Q	R	S
47	8	S	T	R	Q
48	8	T	S	R	Q
49	9	V	U	X	W
50	9	W	U	X	V
51	9	X	V	W	U
52	9	U	W	X	V
53	9	W	X	U	V
54	9	U	X	W	V
55	10	2	Y	Z	1
56	10	Z	Y	2	1
57	10	1	2	Y	Z
58	10	Z	Y	1	2
59	10	Z	2	Y	1

3.4 Kiwifruit Reflectance Estimation using CNN

3.4.1 Network Architecture

A single convolutional layer with three hidden layers was chosen as a starting architecture. This was chosen to keep the network as small and relatively simple as possible. The number and size of layers and their effect on the network will be explored in the results section. A kernel size of 3x3 was chosen as is standard in most CNNs. Stride was set to 1 and padding was set to 0 as a starting point. This resulted in three fully connected layers of size 313632, 21, and 14 (Figure 3.17). RMSE was used as the loss function as many CNNs have used prior (Zhao, et al., 2020; Racek & Baláž, 2010) and the optimiser was set to Adam. Batch size was initially set to 32 as it is default for some training packages (Keras, 2022a). Learning rate was initially set to 0.001 (Keras, 2022b).

Prior to input, all images were scaled to a size of 200 x 200 pixels. This ensures the network is seeing the same parts of fruit in each image. Similarly, to reduce the number of outputs when predicting reflectance, a principal component analysis (PCA) was undertaken on the hyperspectral data.

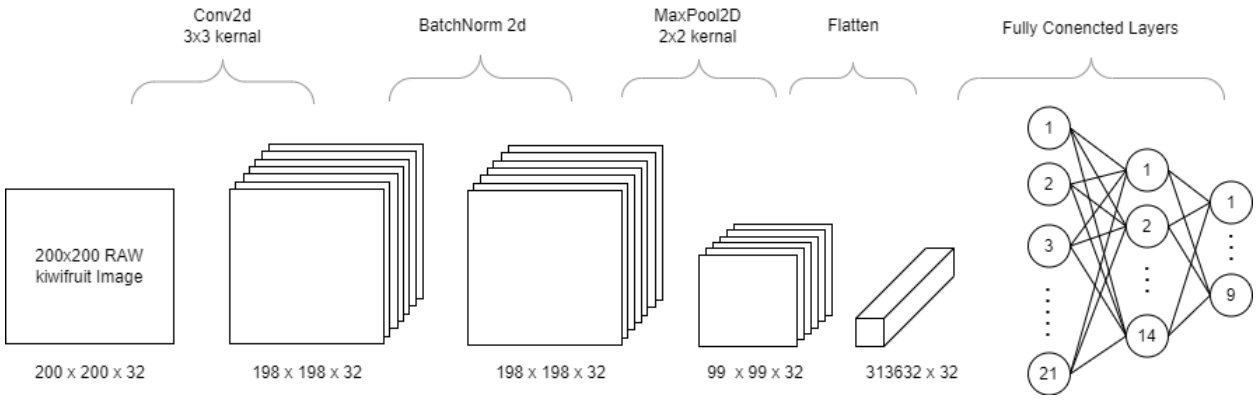


Figure 3.18: Initial CNN architecture where an 200x200px demosaiced kiwifruit image is input into the network, followed by a convolution with a 3x3px kernel. After convolution, the result in normalised and then a Max Pooling operation with a 2x2 kernel is run which shrinks the data size by a factor of two. Finally, the data is flattened into a 1D array and passed through two fully connected layers with the output layer of neurons representing the value of a single principal component relating to the spectral reflectance of the kiwifruit from 400nm – 1000nm. The network is shown with a batch size of 32

This model was implemented using PyTorch.

3.4.2 Evaluation Metrics

As per section 2.6, root mean square error (RMSE) was a widely used error metric for comparing networks. Likewise, the spectral angle mapper (SAM), mean square error (MSE), and mean relative absolute error (MRAE) appear in literature so these metrics will be included (Zhao, et al., 2020; Racek & Baláž, 2010).

The CNN model will be evaluated using mean absolute error (MAE), MSE, RMSE, MRAE, and SAM.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|I_{gt}^{i} - I_{est}^{i}\right|$$

Where I_{gt}^{i} is intensity of the ground truth value and I_{est}^{i} is the intensity of the estimated value.

Equation 3.5: Mean absolute error

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}\left(I_{gt}^{i} - I_{est}^{i}\right)^{2}}{N}}$$

Where I_{gt}^{i} is intensity of the ground truth value and I_{est}^{i} is the intensity of the estimated value.

Equation 3.6: Root mean square error

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(I_{gt}^{i} - I_{est}^{i}\right)^{2}$$

Where I_{gt}^{i} is intensity of the ground truth value and I_{est}^{i} is the intensity of the estimated value.

Equation 3.7: Mean Square Error

$$MRAE = \frac{1}{n}\sum_{i=1}^{n}\frac{\left|I_{gt}^{i} - I_{est}^{i}\right|}{I_{gt}^{i}}$$

Where I_{gt}^{i} is intensity of the ground truth value and I_{est}^{i} is the intensity of the estimated value.

Equation 3.8: Mean Relative Absolute Error

$$SAM = \frac{1}{n}cos^{-1}\frac{\sum_{i=1}^{n} I_{gt}^{i}\, I_{est}^{i}}{\sqrt{\sum_{i=1}^{n} {I_{gt}^{i}}^{2}}\sqrt{\sum_{i=1}^{n} {I_{rec}^{i}}^{2}}}$$

Where I_{gt}^{i} is intensity of the ground truth value and I_{est}^{i} is the intensity of the estimated value.

Equation 3.9: Signal angle mapper

3.4.3 Estimation of Camera RAW data

In keeping with the findings of Stanikunas et al., (2004), this work chose to train the model using five different illuminants. However, the logistics of capturing images of highly perishable samples under five illuminants in a controlled manner on many smartphones is challenging. To overcome this issue, each kiwifruit's spectral reflectance data was used alongside a database of smartphone spectral sensitivities (Jiang et al., 2013) and the standard illuminants D50, D65, A, F2, and the halogen light of the HIS system to estimate a series of RAW images for each camera's unique colour space. Considering equation 2.9 for the iPhone 6, under D65 Illuminant, over the visible range, assuming a viewing angle of 0°, and where $k(\gamma)$ is assumed to be included in the measured smartphone sensitivity, the intensity of a given pixel is,

$$I_C = \int_{400\text{nm}}^{780\text{nm}} S_{iPhone6}(\lambda) * L_{D65}(\lambda) * F(\lambda, 0°)\ d(\lambda)$$

Equation 3.10: Pixel value from iPhone 6 under D65 illuminant a 0° viewing angle

Code was written in MATLAB to compute these images and can be found in Appendix 8.5.

3.4.3.1 Validation of methodology

To explore the methodology of constructing sRGB images from reflectance data, a reflectance image of the X-Rite ColorChecker was captured using the same HS system as the kiwifruit images. The illuminant was captured from the white calibration tile attached to the HS rig using the HS Camera. An average reflectance of each tile was then calculated, see figures 3.19 through 3.22.

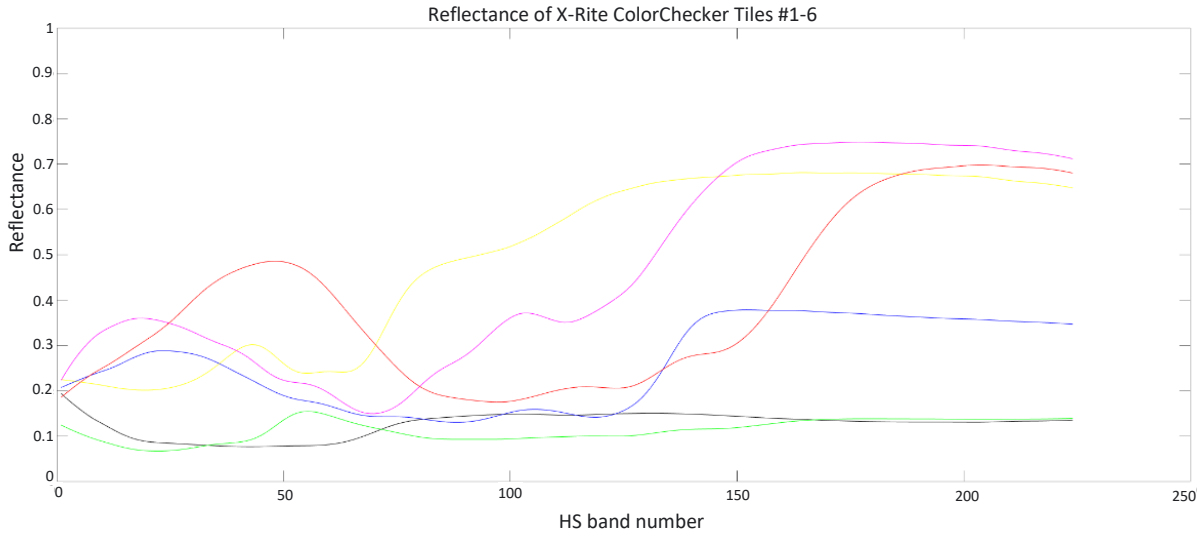


Figure 3.19: X-Rite ColorChecker Reflectance tiles 1-6, dark skin(black), light skin (yellow), blue sky (blue), foliage (green), blue flower (magenta), bluish green (red).

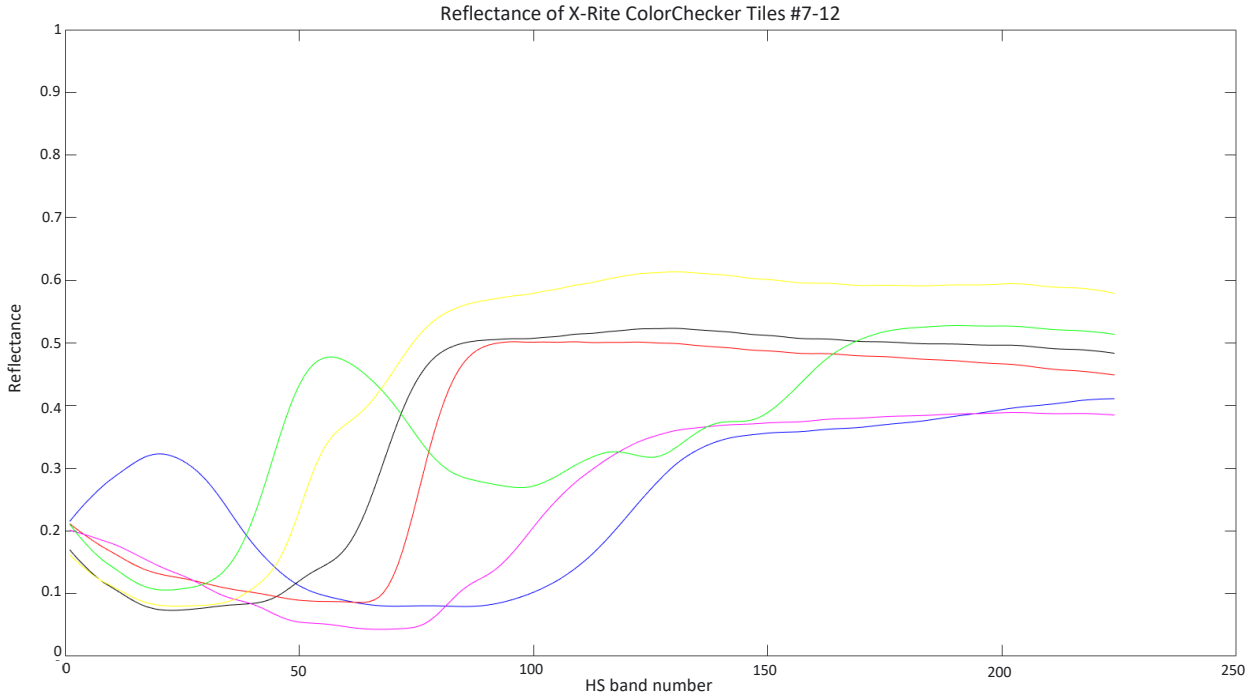


Figure 3.20: X-Rite ColorChecker Reflectance tiles 7-12, orange (black), purplish blue (blue), moderate red (red), purple (magenta), yellow green (green), orange yellow (yellow)

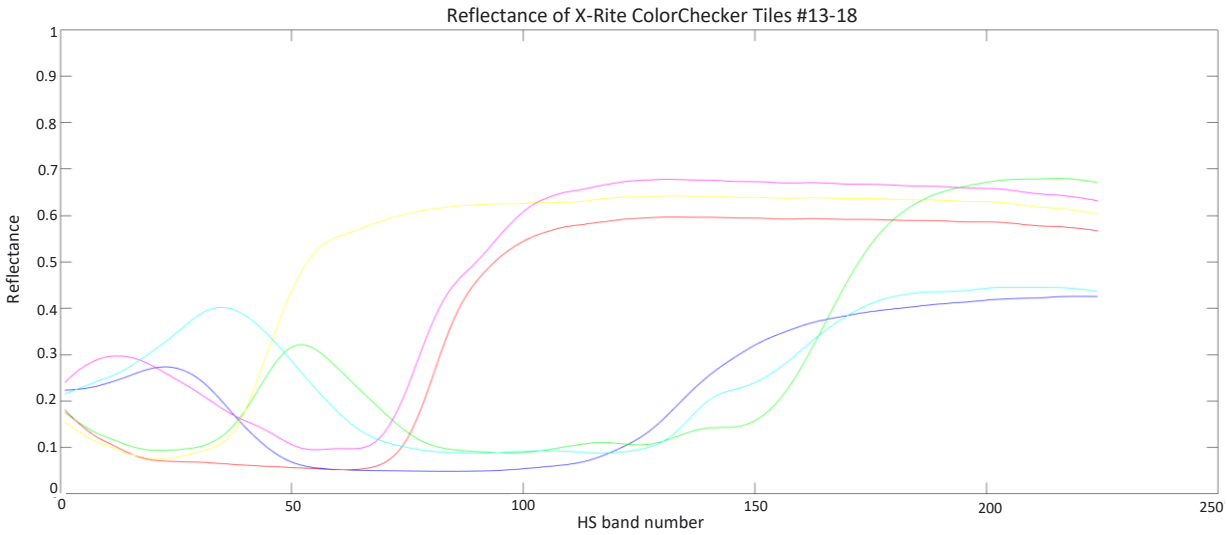


Figure 3.21: X-Rite ColorChecker Reflectance tiles 7-12, blue, green, red, yellow, magenta, cyan

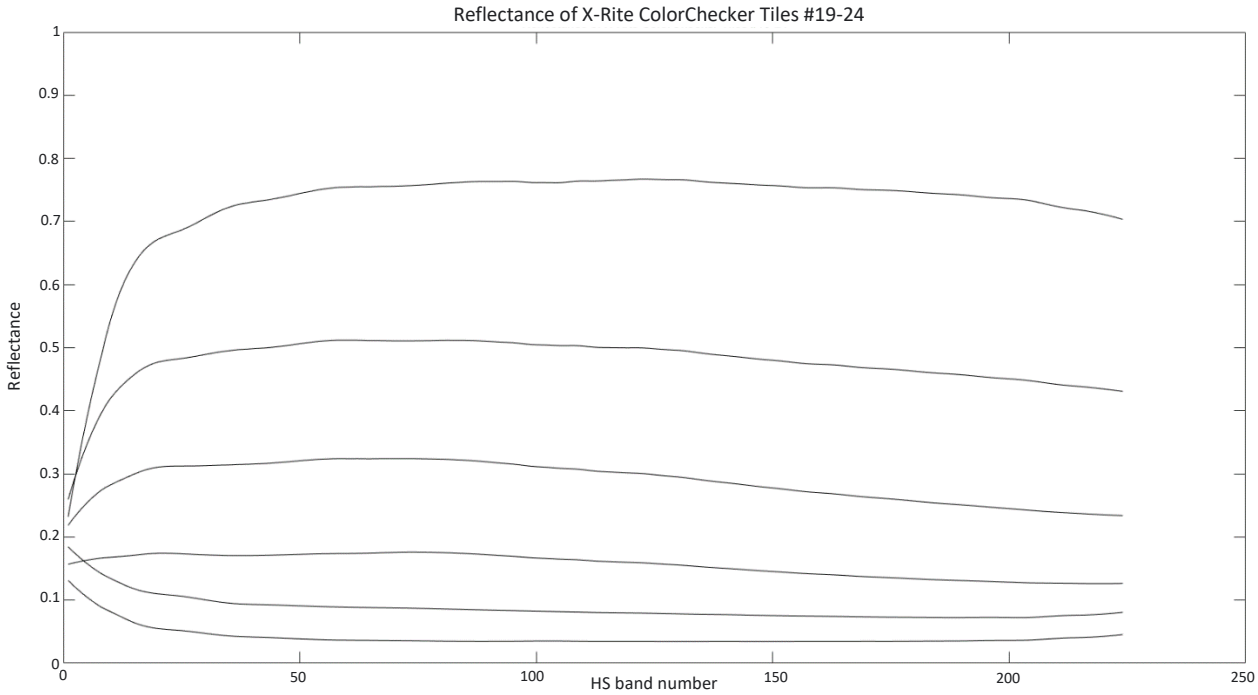


Figure 3.22: X-Rite ColorChecker Reflectance tiles 19-24, white, neutral 8, neutral 6.5, neutral 5, neutral 3.5, and black

The reflectance image was then projected into sRGB using the CIE 1931 standard colourmetric observer and the D65 illuminant and the sRGB values were compared to the ground truth values provided by X-Rite (X-Rite Incorporated, 2009). The projection (Figure 3.22) is shown below.

(A) (B)

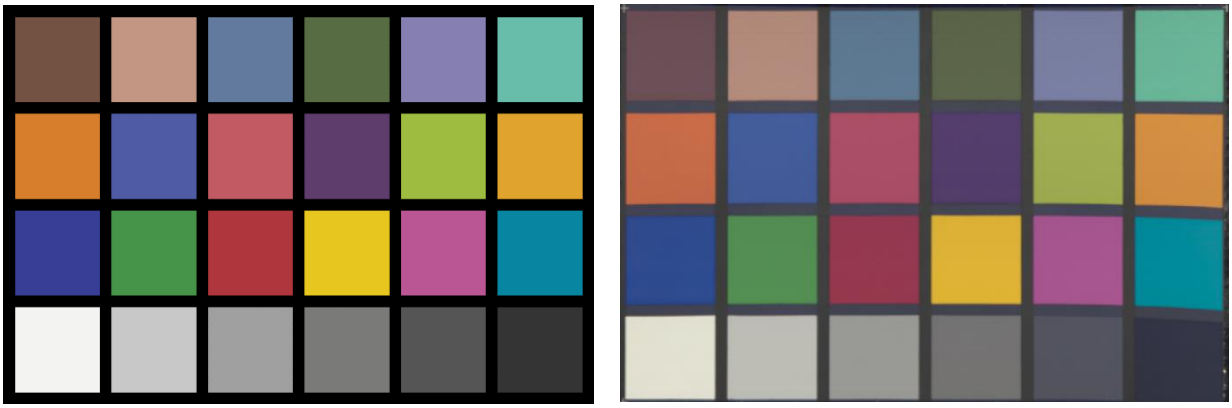


Figure 3.23: Comparison of computer generated (A) (Image Analyst, 2022) and estimated (B) X-Rite ColorChecker

The mean error for the three RGB channels is 8.71, 6.08, and -1.17 respectively with standard deviations of 9.08, 8.77, and 16.33. There are large errors, particularly on the blue channel.

To ensure we are projecting from HS to sRGB correctly the reflectance data captured from (The BabelColor Company, 2022) was used to generate sRGB values using our method as well as that provided by (Magnusson, et al., 2020). The error in each RGB channel is shown in table 3.3.

Table 3.3: Comparison of SRGB errors from The Babel Company, Magnusson, et al. (2020), and this project when using The Babel Company's reflectance data.

	The Babel sRGB			(Magnusson, et al., 2020)			**Our Results**		
sRGB Values	**R**	**R**	**R**	**R**	**G**	**B**	**R**	**G**	**B**
Dark Skin	0	0	0	0	0	0	0	0	0
Light Skin	1	-1	-2	1	-1	-2	1	-1	-2
Blue Sky	-5	1	0	-5	1	0	-5	1	0
Foliage	4	0	-2	4	0	-2	4	0	-2
Blue Flower	-3	1	-2	-3	1	-2	-3	1	-2
Bluish Green	-5	2	0	-5	2	0	-5	2	0
Orange	6	-3	2	6	-3	2	6	-3	2
Purplish Blue	-8	1	2	-8	1	2	-8	1	2
Moderate Red	1	-6	-2	1	-6	-2	1	-6	-2
Purple	-3	-1	-4	-3	-1	-4	-3	-1	-4
Yellow Green	4	1	-2	4	1	-2	4	1	-2
Orange yellow	5	-2	-6	5	-2	-6	5	-2	-6
Blue	-14	2	-3	-14	2	-3	-14	2	-3
Green	2	1	-1	2	1	-1	2	1	-1
Red	0	-4	-3	0	-4	-3	0	-4	-3
Yellow	7	1	-10	7	1	-10	7	1	-10
Magenta	1	-2	1	1	-2	1	1	-2	1
Cyan	-8	4	5	-8	4	5	-8	4	5
White	2	2	-2	2	2	-2	2	2	-2
Neutral 8	1	2	1	1	2	1	1	2	1
Neutral 6.5	1	2	1	1	2	1	1	2	1
Neutral 5	-2	-1	0	-2	-1	0	-2	-1	0
Neutral 3.5	-2	0	0	-2	0	0	-2	0	0
Black	-2	-2	-1	-2	-2	-1	-2	-2	-1
Average Error	-0.71	-0.08	-1.17	-0.71	-0.08	-1.17	-0.71	-0.08	-1.17
STD Error	4.88	2.26	2.94	4.77	2.22	2.88	4.77	2.22	2.88

Table 3.3 shows that both our method and (Magnusson, et al., 2020) generate the same error thus the HS to sRGB process is working correctly. Next, the reflectance data was compared to that measured by Inoue et al., (2017) (Figure 3.23). It appears that there is an error between the bands 0 – 25 in the reflectance data which corresponds the wavelengths between 400nm - 450nm.

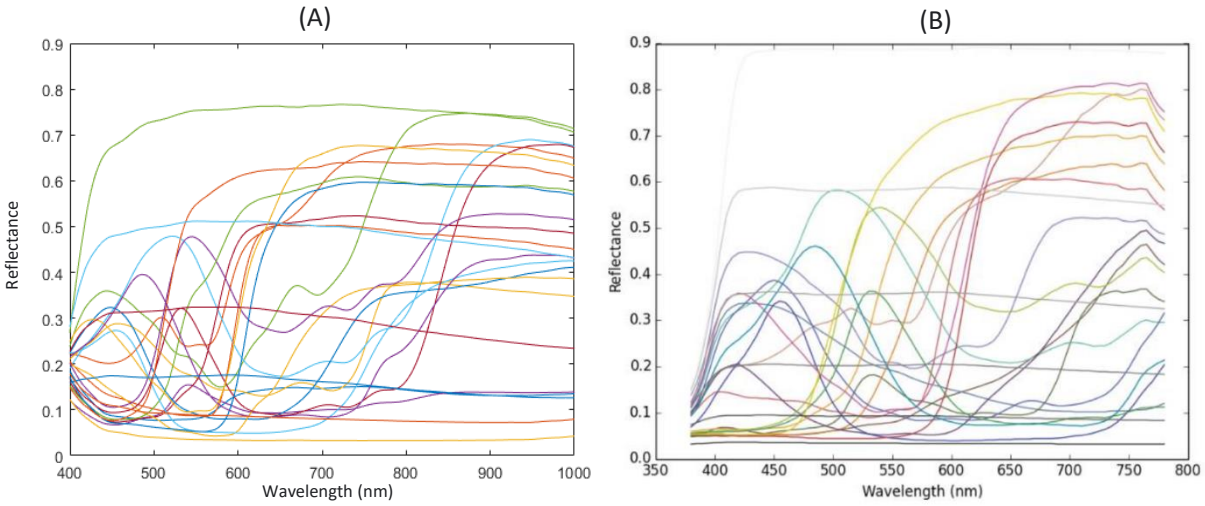


Figure 3.24: Comparison of this project's reflectance values for the 24 ColorChecker tiles (A) vs Inoue et al., (2017) (B)

Visualising the reflectance image (Figure 3.24), this appears to be due to instrument noise in the lower wavelength range of the instrument. Figure 3.24 shows a distinct grainy appearance in the images representing the lower wavelengths. Where it is expected that the white and black tiles of the CCT would have distinctly different intensities, there is little difference in our data.

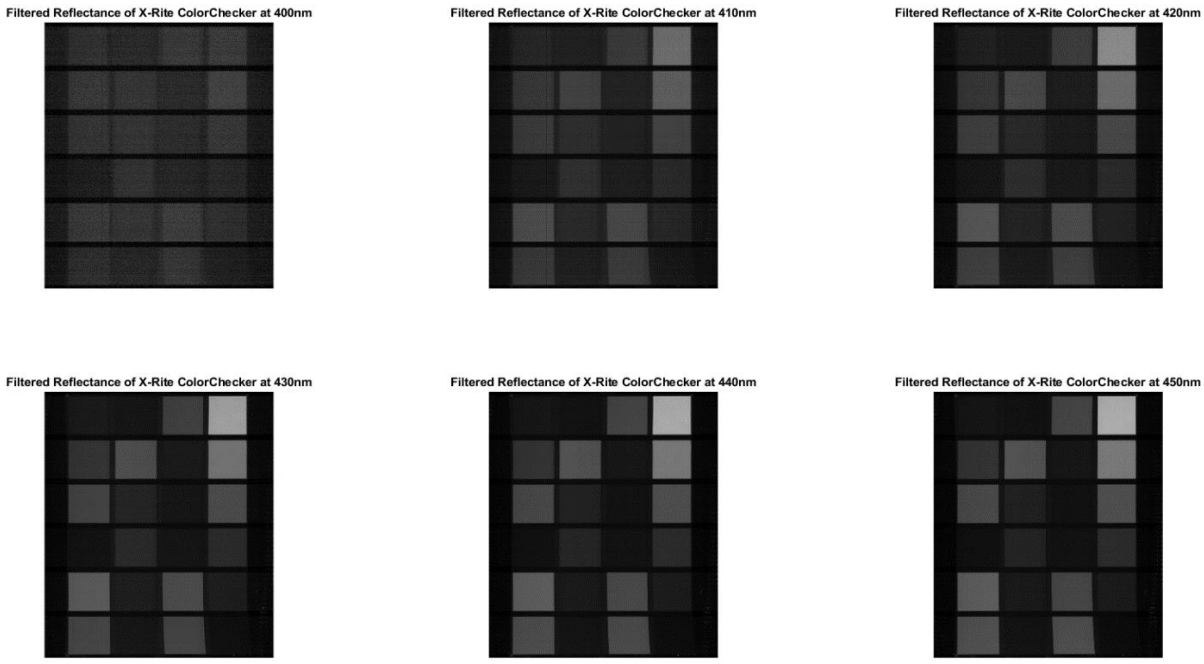


Figure 3.25: Comparison of reflectance of ColorChecker between the ranges of 400nm to 450nm

Using the reflectance data from The Babel Company, our reflectance data within the range of 400nm – 450nm was augmented by linear interpolation and smoothing. This involved using the data provided from The Babel Company between 400-450nm and our own data from 450-1000nm. These datasets were first concatenated and then linearly interpolated between the boundary of the two datasets, i.e. between 450nm

(Babel) and 453nm (our data). MATLAB's smoothdata function was used with a 5 point window to smooth the result.

Comparison of the resulting sRGB values from the new datasets is shown below (Table 3.4) that compares the sRGB values in 8 bits for D65 illuminant over the wavelengths 380nm to 780nm.

Table 3.4: Comparison of sRGB errors using raw, linearly interpolated, and smoothed data vs The Babel Company

	The Babel sRGB			Ours Filtered			Ours Linear Interpolation			Ours Linear Interpolation + Smoothing		
sRGB Values	**R**	**G**	**B**	**R**	**G**	**B**	**R**	**G**	**B**	**R**	**G**	**B**
Dark Skin	0	0	0	-6	-2	17	4	9	13	4	9	13
Light Skin	1	-1	-2	-11	-11	-5	4	9	9	4	9	9
Blue Sky	-5	1	0	-4	-4	-11	8	17	21	8	17	21
Foliage	4	0	-2	1	-6	7	22	4	8	22	4	8
Blue Flower	-3	1	-2	-7	-4	-16	-1	21	21	-1	21	21
Bluish Green	-5	2	0	-5	-11	-16	31	17	9	31	17	9
Orange	6	-3	2	-13	-14	27	5	-13	23	4	-13	23
Purplish Blue	-8	1	2	-8	-2	-13	-12	23	22	-12	23	22
Moderate Red	1	-6	-2	-15	-10	2	-11	-1	16	-10	-1	16
Purple	-3	-1	-4	-5	0	-1	-6	16	21	-6	16	21
Yellow Green	4	1	-2	-4	-13	19	36	1	0	36	1	0
Orange yellow	5	-2	-6	-13	-16	19	15	-13	17	15	-13	17
Blue	-14	2	-3	-4	10	-7	-31	37	12	-32	37	12
Green	2	1	-1	3	-6	12	43	11	-3	43	10	-3
Red	0	-4	-3	-17	3	18	-18	13	11	-17	13	11
Yellow	7	1	-10	-11	-16	24	25	-8	1	25	-8	1
Magenta	1	-2	1	-11	-1	-8	-8	15	31	-8	15	31
Cyan	-8	4	5	-8	1	-4	-8	34	13	-8	34	13
White	2	2	-2	-17	-18	-31	18	11	23	18	11	23
Neutral 8	1	2	1	-11	-11	-18	18	14	25	18	14	25
Neutral 6.5	1	2	1	-6	-6	-10	17	14	22	17	14	22
Neutral 5	-2	-1	0	-6	-6	-6	11	9	16	11	9	16
Neutral 3.5	-2	0	0	-3	-1	9	8	14	14	8	14	14
Black	-2	-2	-1	-1	0	16	4	12	10	5	11	10
Average Error	-0.71	-0.08	-1.17	-7.58	-6	1	11.29	10.79	15.88	11.29	10.79	16
STD Error	4.88	2.26	2.94	5.31	6.90	15.50	5.38	4.19	3.46	5.36	4.12	3.50

Table 3.4 shows that the blue channel from our reflectance is noisy as expected. It also appears that the interpolation has introduced an offset of ~10 sRGB from the true value although it has significantly reduced the standard deviation of each channel, especially the blue channel. Figure 3.25 and 3.26 shows the interpolation of the Babel company and this project's reflectance on individual ColorChecker tiles.

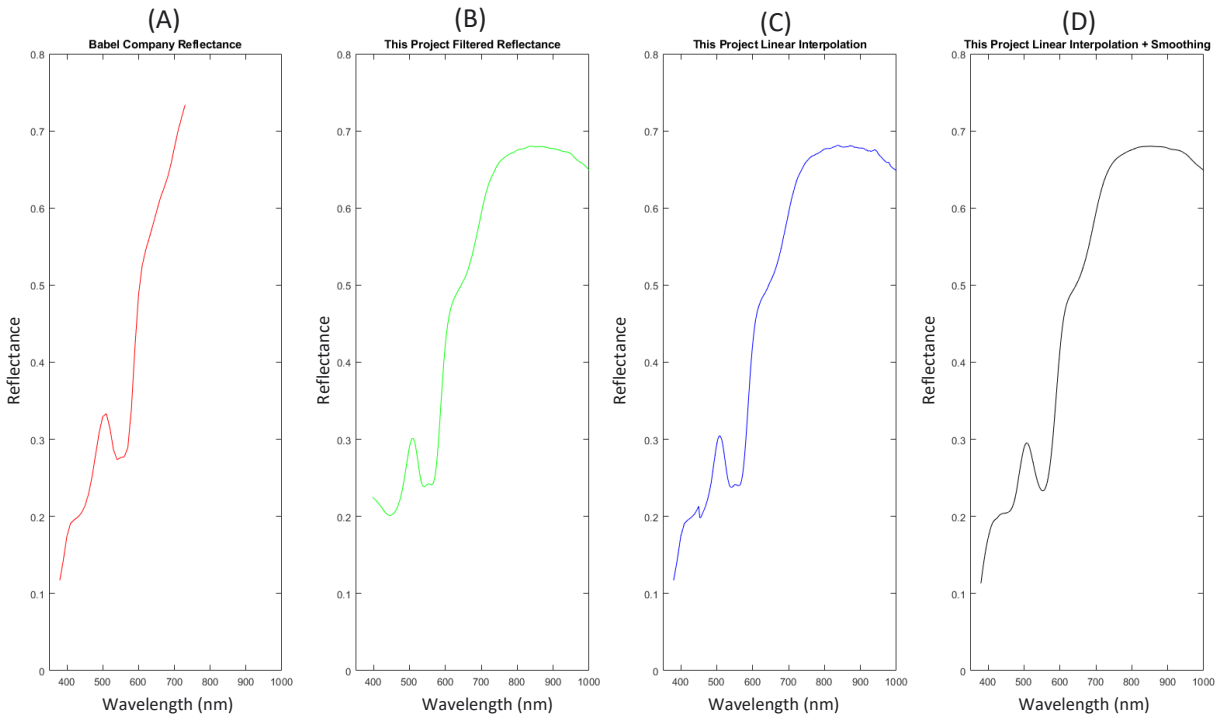


Figure 3.26: Comparison The Babel Company (A), this project's filtered spectra (B), linearly interpolated spectra (C), and linearly interpolated spectra with smoothing (D) of light skin ColorChecker tile.

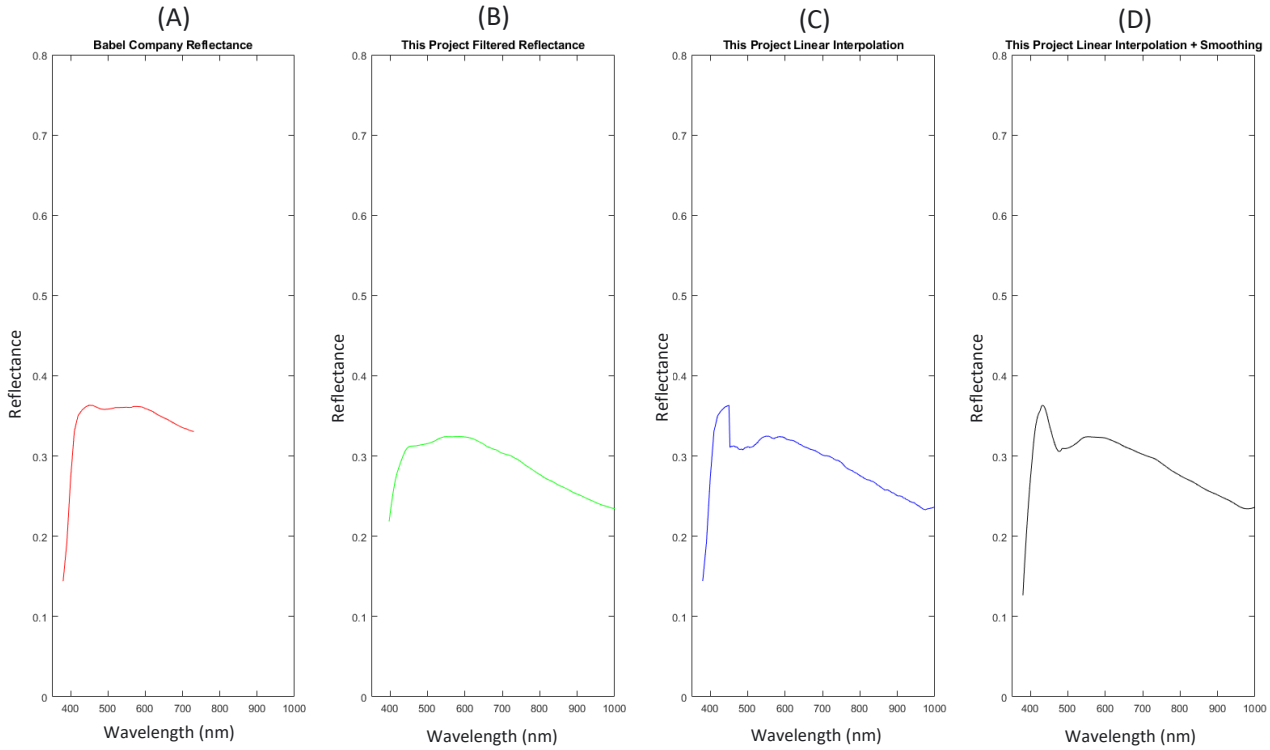


Figure 3.27: Comparison The Babel Company (A), this project's filtered spectra (B), linearly interpolated spectra (C), and linearly interpolated spectra with smoothing (D) of the neutral 6.5 ColorChecker tile spectra

In some cases, particularly the white, grey, and black tiles, there is a large difference in the absolute reflectance values which produce the spikes seen in figure 3.27. To improve this, the range of the Babel data was extended from 380-450nm to 380-500nm. The errors are plotted below in figure 3.27. Individual errors per tile of each method are shown in figures 3.28 through 3.30. These result show that our reflectance data produces noisy sRGB values compared to The BabelColor Company, (2022).

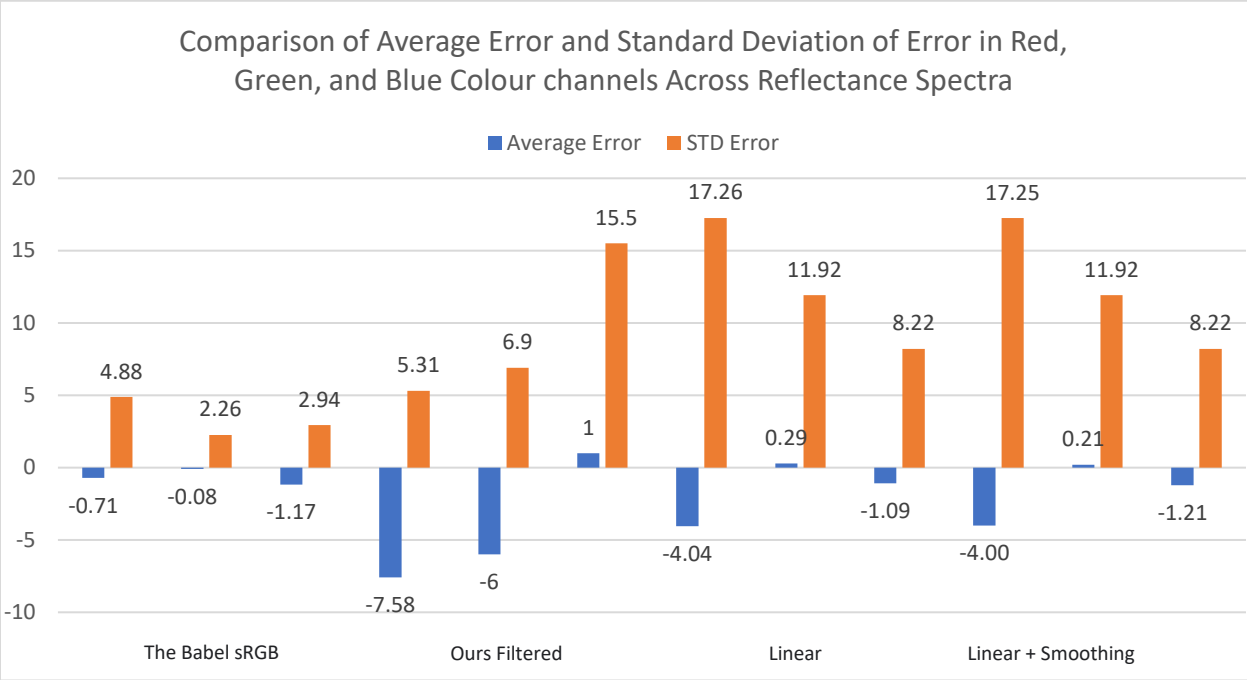


Figure 3.28: Comparison of Average Error (mean error in R,G,B channel across all 24 ColorChecker tiles) and Standard Deviation of Error in red, green, and blue colour channel across reflectance spectra. Because error can be positive and negative, some reflectances produce low average errors, but due to the direction of error, high standard deviations in error.

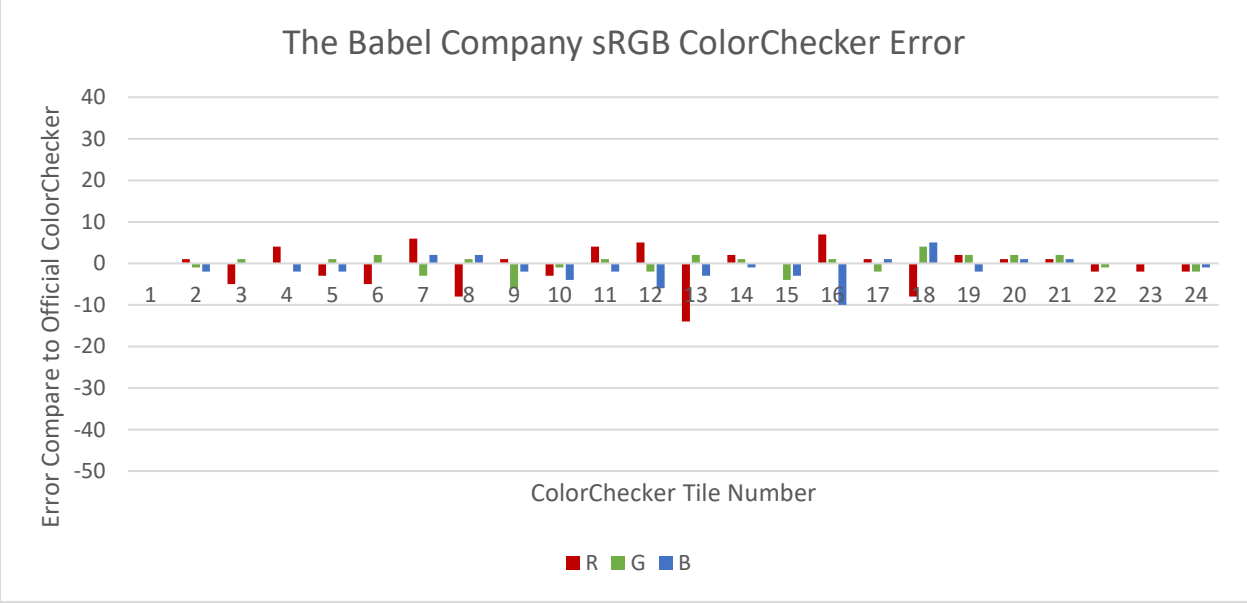


Figure 3.29: The Babel Company sRGB ColorChecker Error by colour channel

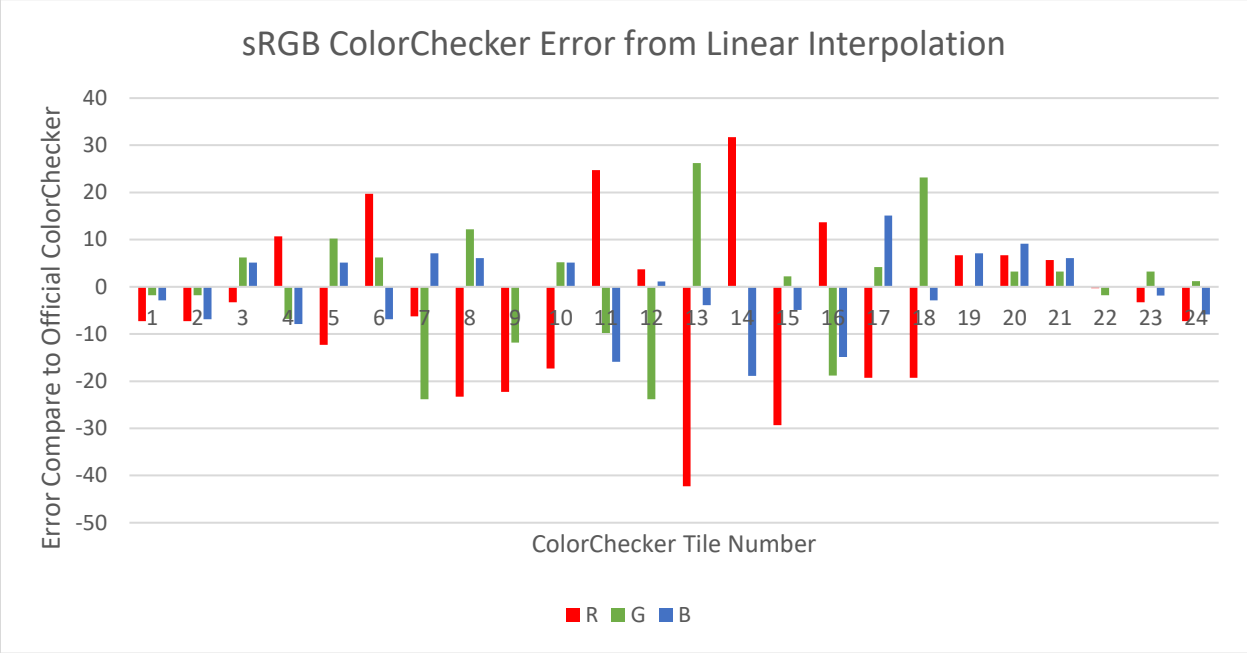


Figure 3.30: sRGB ColorChecker Error from Linear Interpolation by colour channel

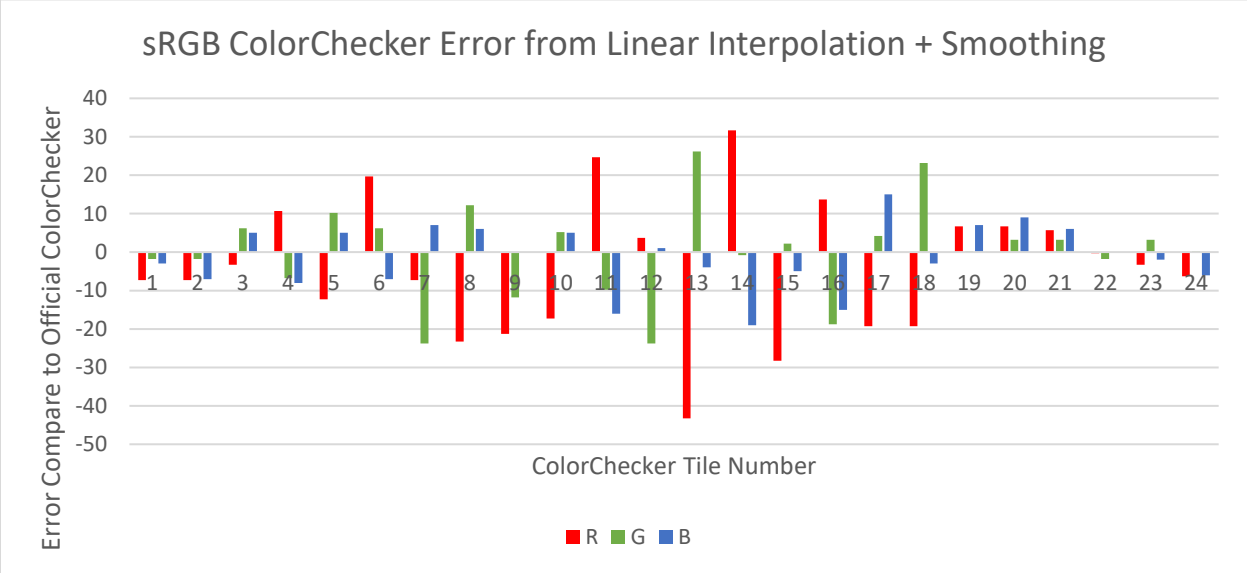


Figure 3.31: sRGB ColorChecker Error from Linear Interpolation + Smoothing by colour channel

The observed offset (Table 3.4) may be explained by this project's reflectance data having lower absolute reflectance values compared to the Babel company's data (Figure 3.31). There is likely some characterisation of the hyperspectral camera, $k(\gamma)$ in equation 2.9, that may also make the reflectance values more accurate. Similarly, the ColorChecker's exact age is unknown – supposedly a few years old – which would also explain why this data has lower amplitudes compared to others as the colours are known to change over time.

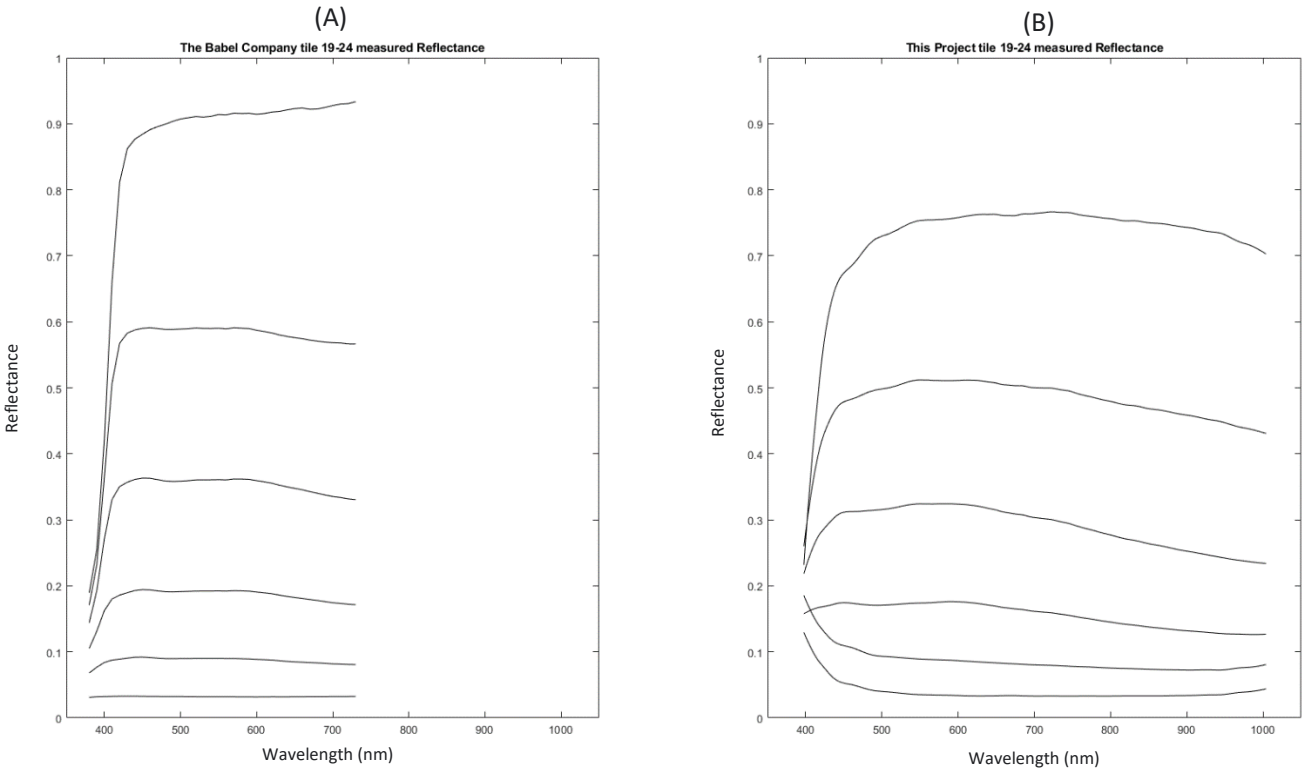


Figure 3.32: Comparison of monochromatic ColorChecker tiles reflectance spectra of The Babel Company (A) vs this project (B)

After using the 380-500nm wavelengths from the Babel data the HS to sRGB process produces lower average errors although much higher standard deviations. Thus, we can conclude that the methodology used to generate sRGB/AdobeRGB images from hyperspectral imagery is sound although the blue wavelengths and any projections from this reflectance data will be noisy between the ranges of 400-450nm.

Considering figure 3.33, it appears that the hyperspectral imaging system does not produce high pixel excitations in the lower end of the spectrum, nears its operational limit. The spectral response provided in the FX10's datasheet (Specim Spectral Imaging Ltd, 2021) shows that the instrument has a relatively low spectral response at 400nm but also has a lower relative response between 900-1000nm. This then should produce grainy looking images at both ends of the spectrum. However, images between 400-450nm look much grainier than those between 900-1000nm. Thus, it can be concluded that due to the use of Tungsten-Halogen lights in the imaging system, the combination of this light source emitting very light energy between the wavelengths 400-450nm and the imaging system having a lower spectral response to these wavelengths, produced noisy data between these wavelength ranges.

3.5 Redness Score prediction using GRNN

There are many ways to regress from reflectance data to redness score including typical linear and polynomial regression techniques, non-linear regressions, and, more recently, machine learning approaches. Implementing a machine learning approach to this regression allows the CNN and fully connected/multilayer perception networks to be concatenated and trained together which may work to decrease overall prediction error. General regression neural networks (GRNN) are used in place of more typical linear regression models and produce state-of-the-art results often due to their ability to fit both linear and non-linear datasets (Chen

& Wang, 2018). Similarly, an SVM trained using MATLAB's *fitrsvm* function (The MathWorks Inc, fitrsvm, 2022) will be used to compare results too.

GRNNs typically feature four layers: an input layer, a pattern recognition layer, a summation layer, and an output layer (Chen & Wang, 2018; Zhou, et al., 2019).

3.5.1 Network Architecture

The effect of network size and shape for a GRNN to predict kiwifruit redness score is explored within this study. As a starting point, a four-layer feedforward network was implemented with a structure such that layer 1, $L1$, has N inputs and $L2 = 1.5N$, $L3 = N/2$, and $L4 = 1$. Layer size is defined as a ratio compared to the input layer $L1$ so that when the size of the network inputs changes the proceeding layers will remain in the same ratio. This was done to explore if different inputs provided more predictive power as opposed to how layer size affects the predictive power for the same input. An example architecture for a network with $L1 = 4$ is shown in figure 3.32. This model was implemented using PyTorch.

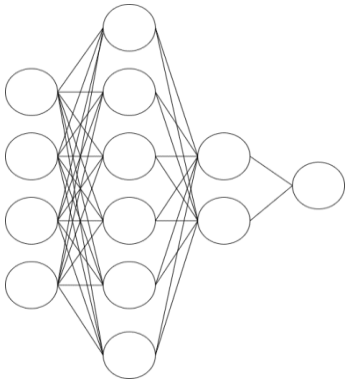


Figure 3.34: Example of GRNN architecture given input layer of size 4

Initial Batch size was set to 64 and initial learning rate to 0.0005. The optimisation of these parameters will be explored later (section 3.5.3). Number of training epochs will be set by observation of initial tests and considering that early stopping offers a way to prevent over-fitting of data. The optimiser used was Adam and the network was trained using RMSE loss function. An initial dropout layer of 20% was used and later optimised. Weight decay was added after some training runs appeared to show signs of exploding gradients. Weight decay was set to 0.001 as higher values prevented the network from converging (Figure 3.33).

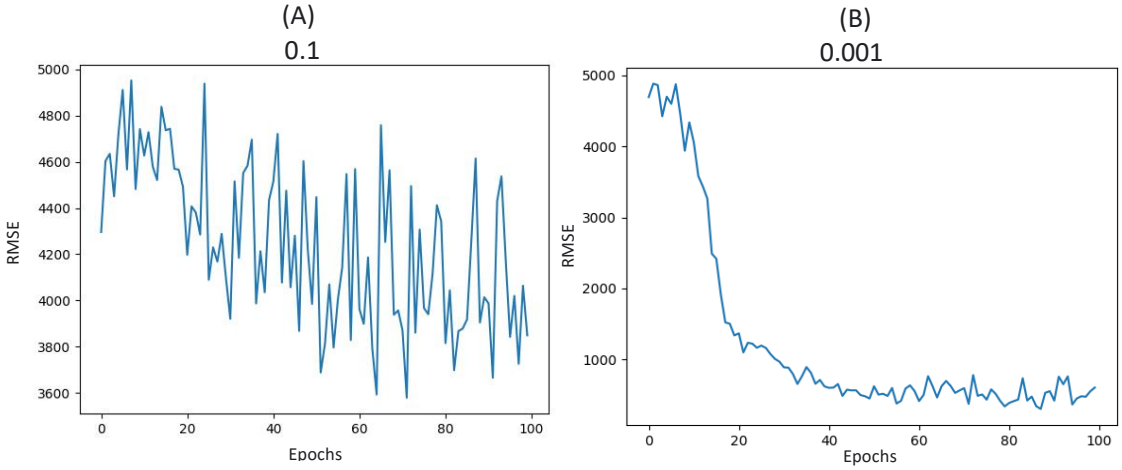


Figure 3.35: Comparison of RMSE error after 100 epochs with weight decay of 0.1 (A) and 0.001 (B)

3.5.2 Evaluation Metrics

RMSE was used as the primary evaluation metric (see section 3.4.2) as it appears commonly in GRNN literature (Chen & Wang, 2018; Zhou, et al., 2019; Stangierski et al., 2019). Similarly, the absolute difference in integrals (AIE) of the estimated and ground truth spectra was used to compare the relative error between growers, phones, and sizes. Recall was used to determine classification accuracy.

$$Absolute\ Integral\ Error = AIE = \int_a^b f(x)\,dx \approx \frac{b-a}{2N}\sum_{i=1}^{N}(f(x_n) + f(x_{n+1}))$$

$$\text{Where } f(x_n) = \left|I_{gt}^i - I_{est}^i\right|$$

Where I_{gt}^i is intensity of the ground truth value and I_{est}^i is the intensity of the estimated value.

Equation 3.11: Absolute Integral Error using Trapz function (The MathWorks Inc, Trapz, 2022)

$$Recall = \frac{True\ Positve}{True\ Positve + False\ Positive}$$

Equation 3.12: Recall

3.5.3 Hyperparameter Tuning

To explore the effect of layer size, optimiser, batch size, input size, dropout percentage, and learning rate, the Weights and Biases (wandb) (Biewald, 2022) hyperparameter tuning package was used. Wandb allows the user to define a set of values for each hyperparameter and then automatically run a series of training cycles to explore the corelation between each hyperparameter and its effect on network error.

A sweep config was initialised using random search and was set to minimise the RMSE training loss of the network. An initial sweep was created to compare the Adam and Stochastic Gradient Descent (SGD) optimisers, visible vs NIR input ranges, dropout percentage between the range of 0.1 and 0.3, and learning rate between the range of 0.0001 and 0.01.

4.0 Results

4.1 Hyperspectral & RAW Images

A total of 1863 hyperspectral images and an accompanying 7452 RAW images were collected on both a Samsung Galaxy S8 and iPhone 12 smartphone using the Adobe Lightroom Application.

All images were uploaded to Google Cloud storage and Adobe CreativeCloud for safe keeping. Spike detection/removal and filtering was competed as per section 3.1.2. Reflectance was calculated as per equation 3.1. Once processed, a total of 1857 reflectance images were useable. Similarly, due to an uploading error, some RAW images were lost during data collection and a total of 7174 images were available for the project.

4.2 Sensory Trial

The key takeaway from the sensory trial is that fruit of 1500 redness score apart appear to show significant agreement of which is redder than the other. Prior to the main study, a suitable starting point regarding difference in redness score was needed. As small test trial was undertaken using a pair comparison with a dataset of fruit in steps of 1000 redness score apart. This was sent online to a 10 colleagues and friends in the building and carried out on their own personal computers at different times of day. This lack of controlled

lighting and viewing conditions should produce larger errors compared to if the test was completed in a controlled environment.

The results showed that when fruit that are 3000 units apart on the ad hoc redness scale are compared, people appear to reach a consensus that fruit of higher redness score are considered perceptually redder. As such, the initial difference in ad hoc redness score was set to 3000 for the main sensory study

Figure 4.1 should be interpreted as follows: each axis represents the redness score of kiwifruit over the range of -2000 – 12000. Each point represents one comparison of two kiwifruit. For example, consider a comparison of two fruit of scores 2000 and 1200, denoted $[2000, 1200]$.

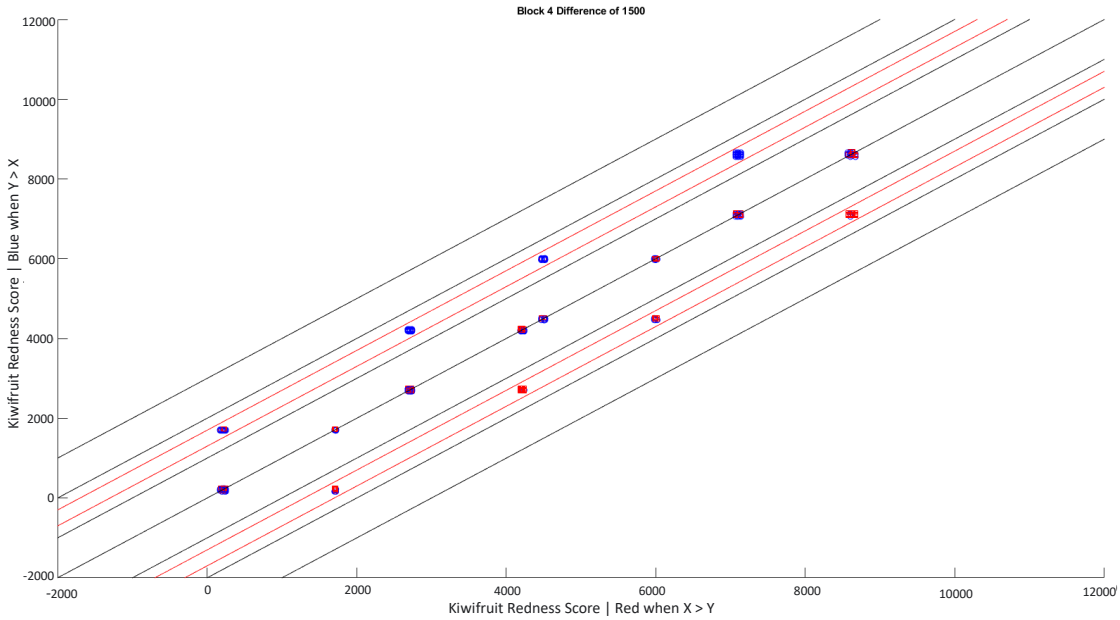


Figure 4.1: Block 4 of sensory trial exploring a difference in redness score of 1500. Red lines denoted the range of valid comparisons that were within 50 of the target redness score (1450 – 1550).

This first fruit score [2000] is the coordinate on the x axis and the other fruit [1200], the y axis. It is expected that the fruit with the higher redness score will be selected over that with the lower score. In this graph, fruit with a higher score on the y axis are denoted with blue points, and points with higher scores on the x axis are denoted with red points. If our scale proved valid, every point that falls above the line $x = y$ should be blue and vice versa for points that fall below $x = y$. Considering our example, $[2000, 1200]$, the fruit of score 2000 is expected to be selected above that of score 1200. Hence, a blue dot will be placed at the coordinate 2000, 1200 on the graph. However, in the case that the lower score is selected by a participant, the point is coloured red and considered to be a point of disagreement. The thresholds of 1500, 1000, and 500 where explored and results are shown in table 4.1 and depicted visually in figures 4.2 through 4.5.

Table 4.1: Average participant agreement vs redness score with metadata

Threshold	Number of Sets	Number of Datapoints	Average Agreement (%)
1500	4	642	92.2115
1000	12	2061	73.4524
500	8	1701	62.0964
0	24	3663	51.1876

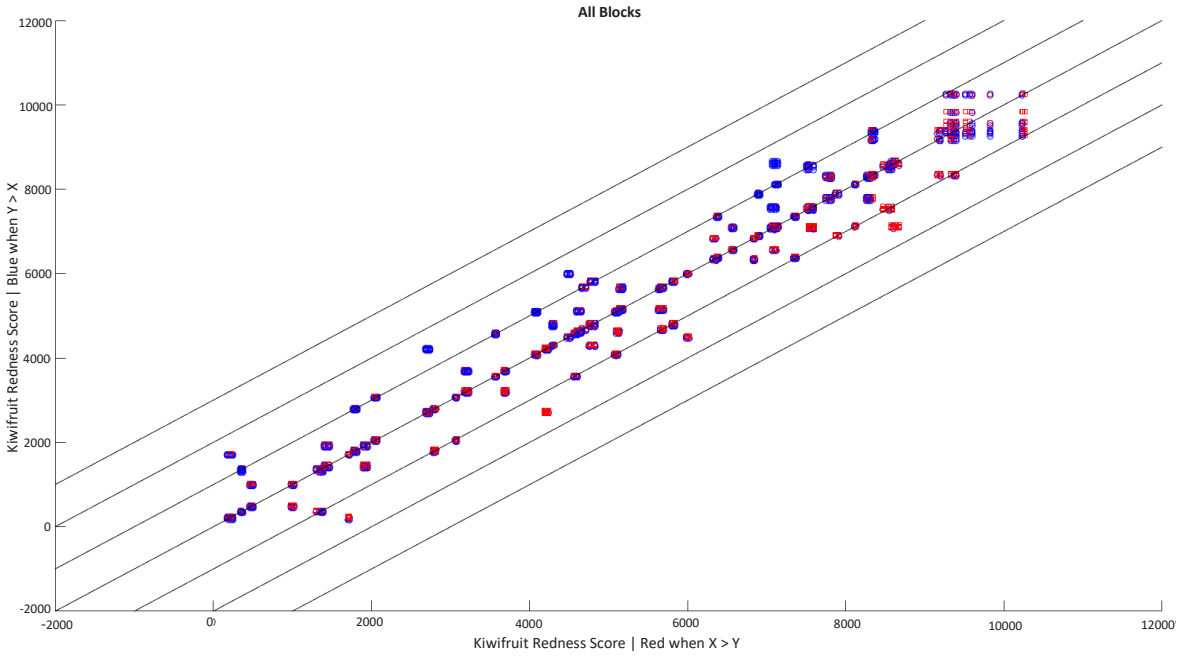


Figure 4.2: Visualisation of all data captured during trial.

Exploring the agreement of participants vs redness score it appears that there is no clear trend under any threshold to suggest participants agree or disagree more or less regarding greener or redder fruit.

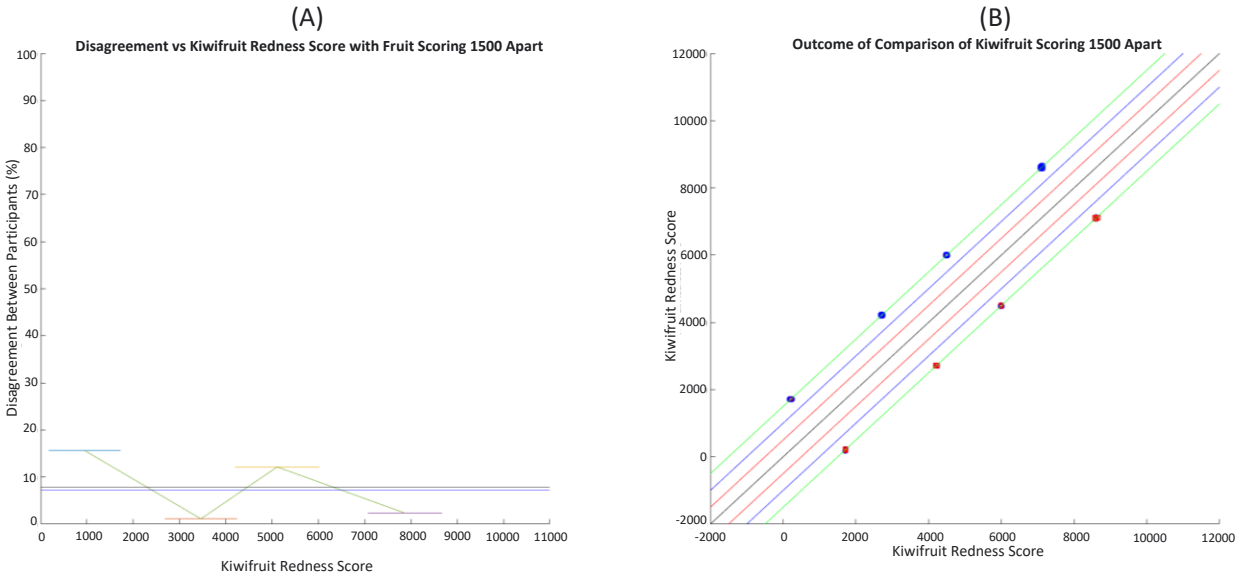


Figure 4.3: Mean and median disagreement over all blocks (A) and visualisation of datapoints (B) when comparing fruit 1500 redness score a part.

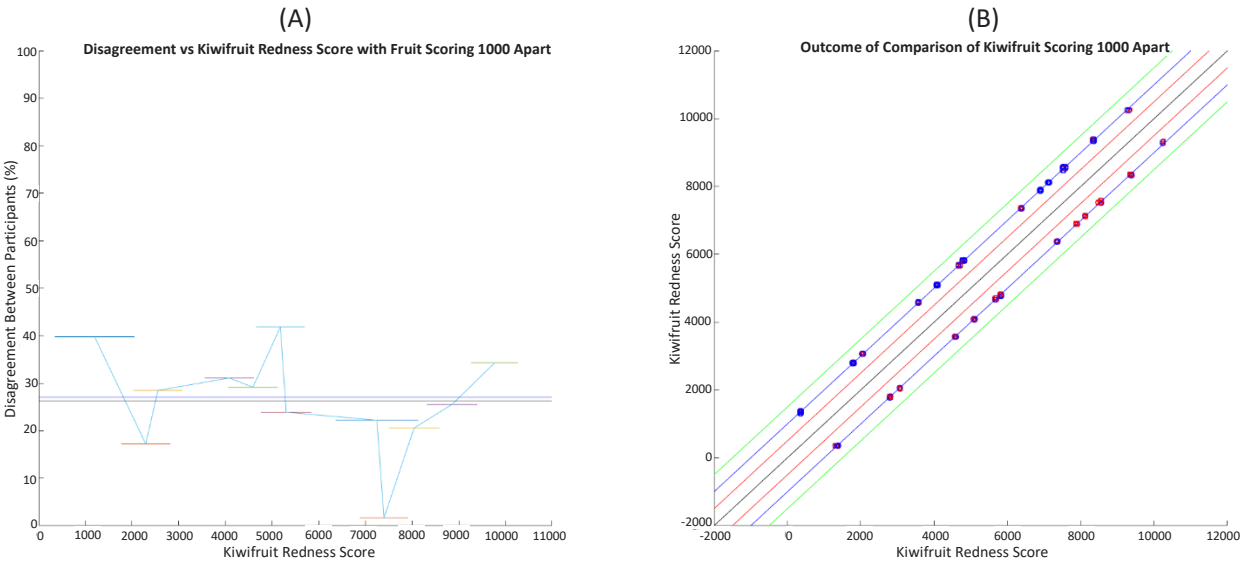


Figure 4.4: Mean and median disagreement over all blocks (A) and visualisation of datapoints (B) when comparing fruit 1000 redness score apart.

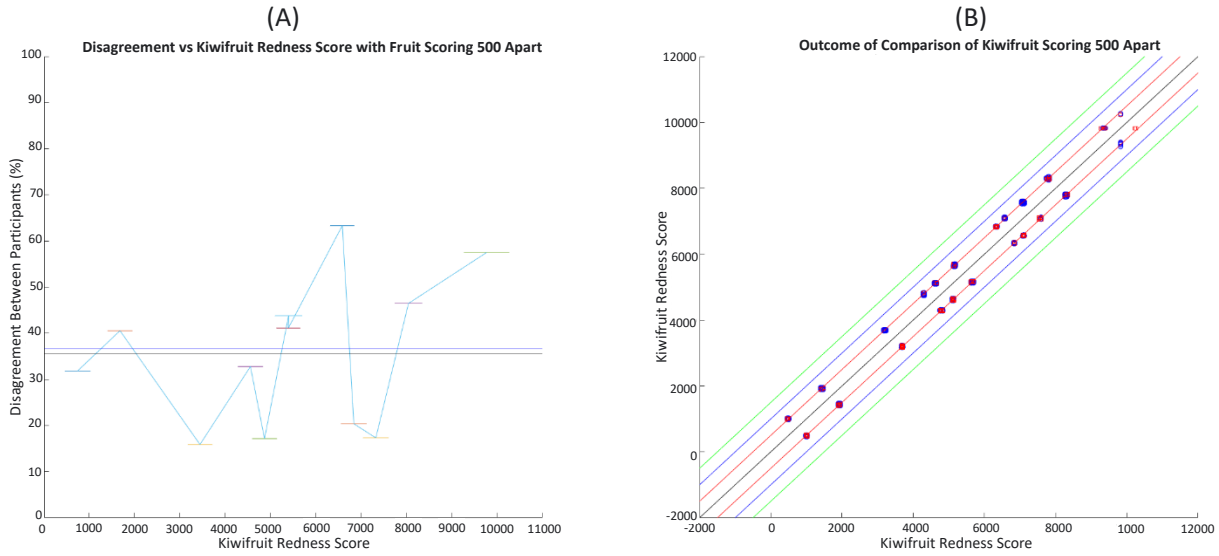


Figure 4.5: Mean and median disagreement over all blocks (A) and visualisation of datapoints (B) when comparing fruit 500 redness score apart.

These results show an expected trend that as the difference in two kiwifruits' redness score reduces, people tend to disagree more often which fruit is redder than another. The above results also appear quite noisy and show quite high variability for different ranges of redness scores. It is likely that whilst this project set out to measure redness as a single dimension, it may not be possible, or at least not using the methods this trial employed, to reduce kiwifruit redness to a single dimension. Regardless a difference in redness score of greater than or equal to 1500 may be used to how often people will agree which kiwifruit is redder than another.

4.3 Kiwifruit Reflectance from RAW images using Spectral Super Resolution

There are two aspects of estimating kiwifruit score from RAW images. Firstly, is it possible to estimate kiwifruit reflectance accurately using RAW images? Secondly, how much reflectance data is needed to successfully score a kiwifruit?

4.3.1 Convolutional Neural Network

To reduce the number of outputs from the network, a principal component analysis (PCA) was undertaken. To consider both the visible (VIS) range (400nm-780nm) alone as well as the visible + near infrared (NIR)(400nm-1000nm) a PCA was computed using 400-700nm and 400-1000nm of the spectrum from each fruit.

Results show that 99.9% of the variance in the visible range and NIR range could be represented by 7 and 9 principal components respectively.

Predicting Average Kiwifruit Reflectance from RAW Images

Taking the RAW data from the Samsung Galaxy S8 and iPhone 12 that had been demosaiced, the network was trained to predict the 7 principal components of each kiwifruit's average reflectance.

An initial comparison was undertaken to explore the effect of using a different loss function on the validation loss. The results (Table 4.2) show that using RMSE as the network's loss function provides the lowest values across all sets of evaluation metrics

Table 4.2: Comparison of network errors when trained using different loss functions

Loss Function	AE	MSE	RMSE
L1	0.0622	0.0083	0.0907
MSE	0.2425	0.1137	0.3370
RMSE	0.0410	0.0039	0.0618

Considering the relative weights of each of principal component (PC) for the visible range, it makes little sense to give them equal weighting in the loss function given the first three PCs greatly outweigh the next four. Table 4.3 shows the relative variance of each of the seven principal components over the visible range.

Table 4.3: Variance explained by each principal component of average kiwifruit reflectance spectra

Principal Component	1	2	3	4	5	6	7	Total
Variance Explained	57.0237	28.9900	12.6931	0.7273	0.3497	0.0707	0.0635	99.9178

An addition to the network's loss was written to weight each estimated PC based on its variance explained (Table 4.4). The results show this did not produce a reduction in loss compared to using no weighting.

Table 4.4: Comparison of network error when weighting the loss function based on the variance of each principal component

RMSE Loss	MAE	MSE	RMSE
Non-Weighted	1.2992	0.0110	0.0273
Weighted	1.4836	0.0128	0.0305

Next the effect of reconstructing the full spectrum for PCs and its effect on network loss was explored (Table 4.5). That is, does the network produce lower errors when using the full spectrum in its loss function. The results show that reconstructing the full spectrum from PCs and then passing it to the loss function significantly reduced the network error and this approach will be continued moving forward.

Table 4.5: Comparison of network errors when calculating loss using principal components vs full spectra

Loss Space	AE	MSE	RMSE	SAM
PCA	0.0410	0.0039	0.0618	0.0273
Full Spectrum	0.0202	0.0002	0.0131	0.0290

Next the loss of the visible and NIR ranges were compared (Table 4.6). It appears that the visible and visible + NIR spectrums produce similar results.

Visible Range vs VIS+NIR

Table 4.6: Comparison of network errors over visible and visible-NIR spectral range

Spectral Range	AE	MSE	RMSE
VIS	0.2024	0.0002	0.0131
VIS-NIR	0.2144	0.0001	0.0113

Exploration of Error by Camera, Angle, Grower, and Size

The next logical step to check that this error is consistent regardless of phone, grower, angle, and size (Table 4.7). It does not appear that any one phone, size, angle, or grower produced significantly higher errors than any others when predicting the average reflectance of a kiwifruit.

Table 4.7: Comparison of network errors by grower, size, phone, and angle over the visible range

	Count	AIE	MRAE	RMSE	SAM
Whole Network	13,002	1.2659 std=0.6125	0.0331 0.0154	0.0118 0.0050	0.0290 0.0110
iPhone	6659	1.2581	0.0331	0.0118	0.0286
Samsung	6343	1.2741	0.0330	0.0118	0.0294
Orthogonal	6464	1.2179	0.0317	0.0114	0.0288
Diagonal	6538	1.3135	0.0344	0.0122	0.0291
Grower – 48785	2470	1.1955	0.0307	0.0110	0.0308
Grower – 27844	2827	1.3068	0.0329	0.0116	0.0292
Grower – 7829e	2710	1.2970	0.0338	0.0123	0.0268
Grower – 5205a	1635	1.4377	0.0391	0.0142	0.0334
Grower – 98948	615	1.1790	0.0327	0.0117	0.0257
Grower – 9205j	0	~	~	~	~
Grower – 7174a	445	1.2923	0.0344	0.0115	0.0300
Grower – 87143	585	1.1282	0.0295	0.0104	0.0279
Grower – 77357	597	1.2395	0.0326	0.0114	0.0275
Grower – 32204	420	1.2577	0.0324	0.0118	0.0299
Grower – 26278	698	1.0249	0.0283	0.0097	0.0235

Size – 25	681	1.2148	0.0309	0.0113	0.0255
Size – 27	1112	1.3588	0.0359	0.0130	0.0280
Size – 30	922	1.1678	0.0315	0.0113	0.0262
Size – 33	1063	1.1717	0.0321	0.0114	0.0275
Size – 36	3937	1.1828	0.0314	0.0110	0.0268
Size – 39	1200	1.1751	0.0312	0.0112	0.0292
Size – 42	2048	1.3805	0.0346	0.0126	0.0319
Size – 46	1695	1.4339	0.0355	0.0127	0.0343
Size – 49	344	1.3794	0.0394	0.0136	0.0322

It is also possible to assess the network by its ability to predict kiwifruit score or kiwifruit class from demosaiced images. That is, does explicitly or implicitly estimating kiwifruit reflectance lower the networks failure rate?

To simplify the redness scale of kiwifruit, fruit that fell below a score of 0 were clipped to 0 and those above 10,000 were clipped to 10,000. Likewise, to establish classes of fruit, the redness scale was split evenly into five classes based on each class being 2000 score from the next class. Hence, five classes of 0-2K, 2-4K, 4-6K, 6-8K, and 8-10K were created and denoted from 1 (low score) to 5 (high score).

Predicting Kiwifruit Redness Score from RAW images

Using the same network architecture with a single output value, the network produced a mean error, defined as the average difference of the predicted redness score from the ground truth values, of 270.30 with a standard deviation of 196.42 (Table 4.8). If we analyse our features we can see that phone, camera angle, and size do not appear to affect the rate of failure. There are a couple growers who have higher fail rates. This can likely be solved with some data augmentation. Translating this error into a classification failure rate by counting the number of times this error in redness score causes be counted as different than its actual classification, the network successfully classifies kiwifruit 89.5% of the time.

Table 4.8: Comparison of redness score error produce by CNN

Feature	Count	Class Fail Rate	Mean Dist.	Max	Std
Total	11,820	0.01149	270.30	3397.6	196.42
iPhone	5820	0.1005	247.62	2869.9	228.50
Samsung	6000	0.1288	292.30	3397.6	262.97
Orthogonal	5979	0.1162	265.49	2868.9	240.88
Diagonal	5841	0.1135	275.22	3397.6	254.22
Size - 25	424	0.1156	251.32	1238.5	206.89
Size – 27	1097	0.1376	344.47	3397.6	444.74
Size – 30	747	0.1165	266.86	1543.6	235.94
Size – 33	854	0.1838	249.58	1145.5	193.83
Size – 36	3757	0.1038	257.30	1388.3	203.37
Size – 39	1088	0.0873	238.54	1116.9	189.93
Size – 42	1962	0.1075	275.12	1617.54	235.58
Size – 46	1601	0.1205	288.58	1584.7	240.01
Size – 49	290	0.0862	235.17	1181.3	198.81
Grower - 48785	2428	0.1429	230.61	997.47	175.96
Grower – 27844	2400	0.1271	306.41	1617.5	263.86

Grower – 7829e	2301	0.1182	276.42	3397.6	333.26
Grower – 5205a	1622	0.0692	285.01	1584.7	265.95
Grower – 98948	598	0.0719	261.12	969.79	203.93
Grower – 9205j	0	~	~	~	~
Grower – 7174a	389	0.1388	270.25	1025.7	211.29
Grower – 87143	603	0.0514	223.68	878.26	171.91
Grower – 77357	532	0.1504	270.67	1070.5	211.88
Grower – 33204	387	0.1008	215.52	818.34	163.58
Grower – 26278	560	0.1518	312.69	1191.8	238.83

Similarly, figure 4.6 shows the number of times a class boundary is crossed and at which boundary. There are naturally fewer crossings at the higher boundaries, and this is simply because there is less fruit with higher redness scores, see figure 4.7. Also worth noting, is that misclassified fruit were only one class from their correct value in 99.4% of cases. There were no misclassifications larger than two classes in error.

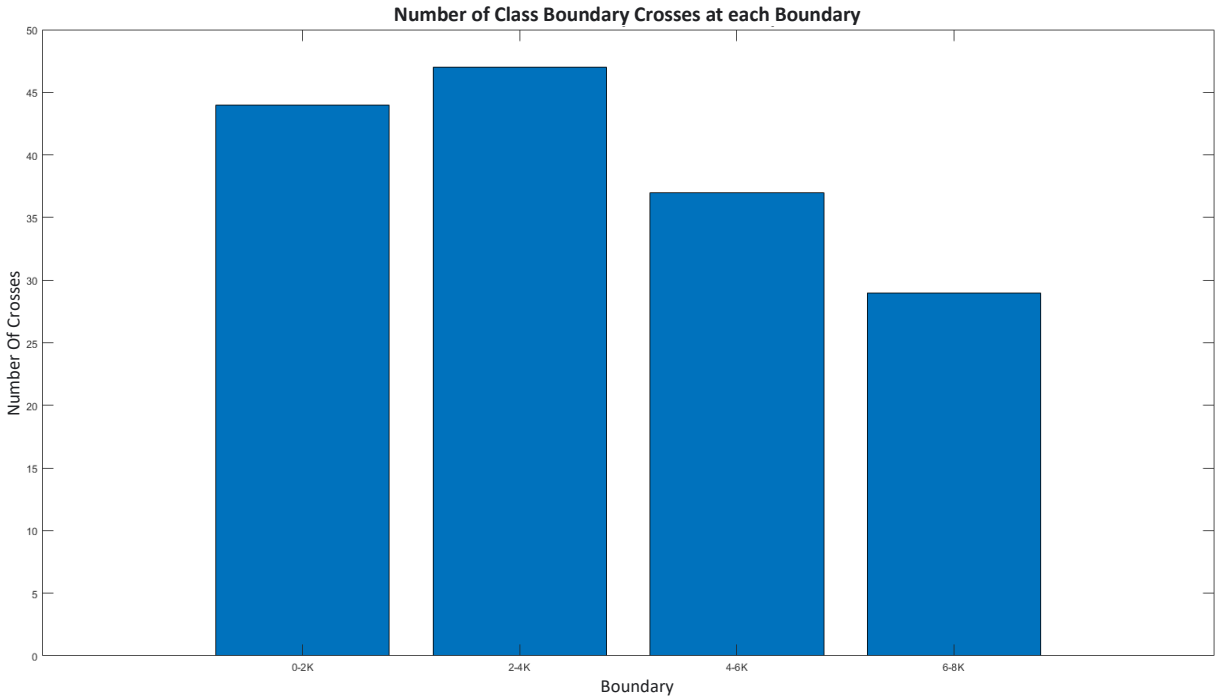


Figure 4.6: Comparison of number of classification errors at each class boundary

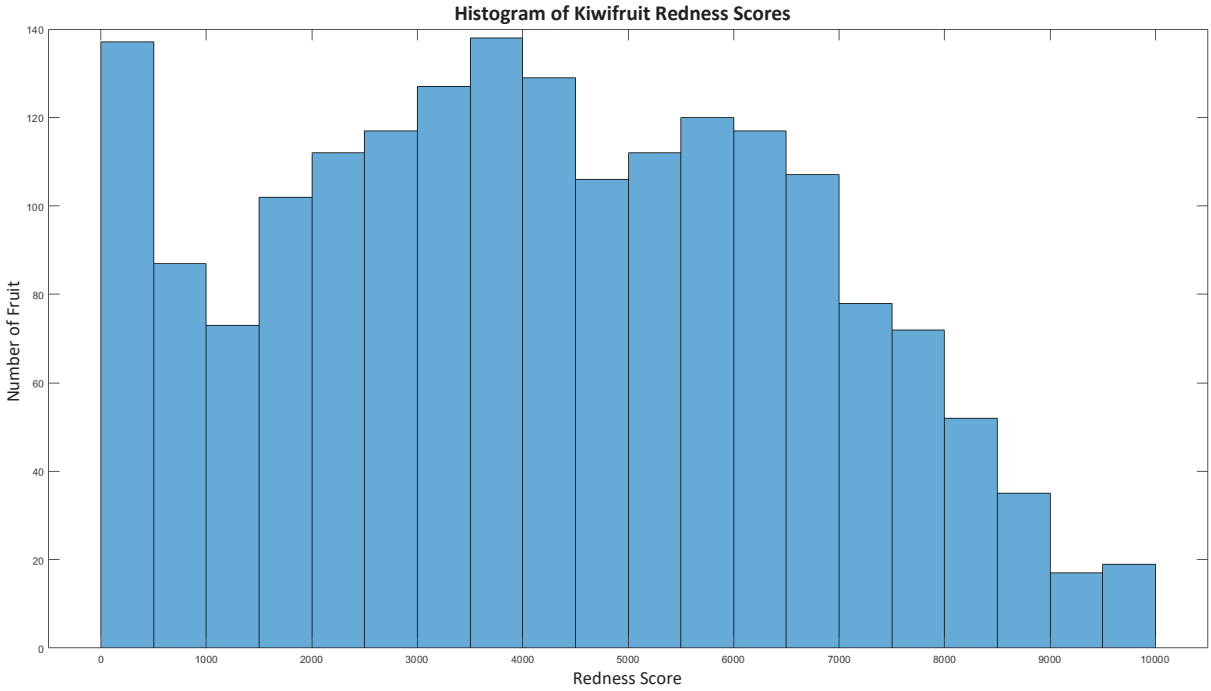


Figure 4.7: Distribution of Kiwifruit redness scores over the whole dataset

Predicting Kiwifruit Class from RAW images

When predicting kiwifruit class from RAW images the network successfully classifies images 91% of the time (Table 4.9). The failure rates are constant for all cameras, sizes, and angles. There is one grower that shows a significantly higher failure rate - between triple and quadruple - compared to the other growers. Visually inspecting this grower, there does not appear anything visually different in their fruit when compared to other growers' fruit.

Table 4.9: Comparison of classification error by grower, size, angle, and phone

Feature	Count	Correct	Efc. Fail Rate
Whole Network	31,914	28,739	0.0905
iPhone	16,314	14,735	0.0968
Samsung	15,600	14,004	0.1023
Orthogonal	16,530	14,966	0.0946
Diagonal	15,384	13,773	0.1047
Size - 25	1604	1427	0.1103
Size – 27	3086	2830	0.0830
Size – 30	2535	2345	0.0750
Size – 33	2713	2437	0.1017
Size – 36	9398	8449	0.1010
Size – 39	2720	2549	0.0629
Size – 42	4680	4136	0.1162
Size – 46	0	~	~
Size – 49	736	671	0.0883
Grower - 48785	6212	5554	0.1059
Grower – 27844	7347	6530	0.1112

Grower – 7829e	6810	3107	0.1032
Grower – 5205a	3946	3720	0.0573
Grower – 98948	1458	1364	0.0645
Grower – 9205j	0	~	~
Grower – 7174a	803	605	0.2466
Grower – 87143	1282	1184	0.0764
Grower – 77357	1258	1152	0.0843
Grower – 33204	1234	1157	0.0624
Grower – 26278	0	~	~

Higher errors appear to be associated with growers with lower fruit counts, it may be that the network does not have enough samples of these growers to learn their nuances. Thus, it was decided to augment grower 7174a's datasets and to see if this alleviated the issue.

Data was augmented in two ways. One augmented dataset was created by rotating the original images through 180°. The other dataset was created by horizontally flipping the dataset. This added an additional 304 images belonging to grower 7174a (Figure 4.8).

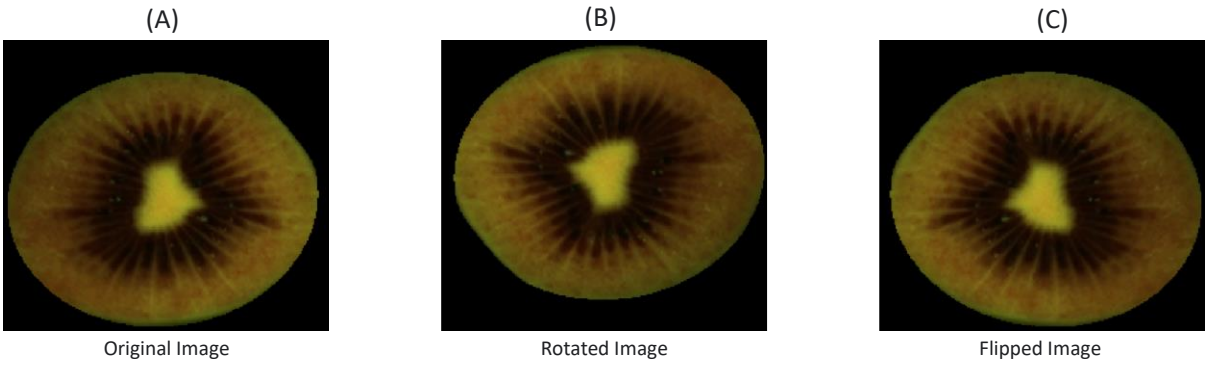


Figure 4.8: Comparison of original kiwifruit image (A) vs fruit when rotated 180° (B) and laterally inverted (C)

After retraining the network, a small reduction in failure rate of grower 7174a can be seen (Table 4.10).

Table 4.10: Network classification error after retraining with augmented data

Grower	Original Fail Rate	Augmented Fail Rate	Change
Grower - 48785	0.1059	0.0919	-0.0140
Grower – 27844	0.1112	0.0635	-0.0477
Grower – 7829e	0.1032	0.0982	-0.0050
Grower – 5205a	0.0573	0.0362	-0.0211
Grower – 98948	0.0645	0.0852	+0.0207
Grower – 9205j	~	~	~
Grower – 7174a	0.2466	0.2302	-0.0164
Grower – 87143	0.0764	0.0550	-0.0214
Grower – 77357	0.0843	0.1306	+0.0463
Grower – 33204	0.0624	0.0871	+0.0247
Grower – 26278	~	~	~
Network Total	0.0995	0.0918	-0.0076

Whilst it appears that we have decreased the failure rate slightly for grower 7174a, it is still significantly higher than the other growers. This could mean that there is something in grower 7174a's dataset that makes it difficult to estimate the fruit's class, or that many of this grower's fruit fall near a class boundary making it more likely to fail, or the network's ability to estimate kiwifruit class may be affected by the augmentation methods used. In the case of there not being enough data to train for the effects of augmentation, we can add additional data from other growers with lower fruit counts to offset this effect.

The augmented data has a much higher than average failure rate compared to the whole network suggesting the network has not learned invariance to the augmentation procedures (Table 4.11)

Table 4.11: Comparison of classification error of augmented images

Feature	Classification Failure Rate
Whole Network	0.0918
Rotate Images	0.1587
Flipped Images	0.2680

An additional 262 images were augmented from grower 98948 and the network retrained (Table 4.12). The results show a reduction in grower 7174a's misclassification by ~5%.

Table 4.12: Comparison of classification error using a different number of augmented images

Grower	Original Fail Rate	Augmented 7174a	Augmented 7174a & 98948	Change from 7174a
Grower - 48785	0.1059	0.0919	0.0939	+0.0020
Grower – 27844	0.1112	0.0635	0.0673	+0.0038
Grower – 7829e	0.1032	0.0982	0.0861	-0.0121
Grower – 5205a	0.0573	0.0362	0.0342	-0.0020
Grower – 98948	0.0645	0.0852	0.0496	-0.0356
Grower – 9205j	~	~	~	~
Grower – 7174a	0.2466	0.2302	0.1737	-0.0565
Grower – 87143	0.0764	0.0550	0.1086	+0.0536
Grower – 77357	0.0843	0.1306	0.0754	-0.0552
Grower – 33204	0.0624	0.0871	0.0847	-0.0024
Grower – 26278	~	~	~	~
Network Total	0.0995	0.9185	0.0783	-0.0136

After retraining, it can now be seen that the network has learned invariance to the augmented data types (Table 4.13).

Table 4.13: Comparison classification errors after adding additional augmented data from two growers to the network

Feature	Classification Failure Rate
Whole Network	0.0783
Rotated Images	0.1280
Flipped Images	0.0578

We can see that the addition of more augmented data has had a positive impact on both the failure rate of grower 7174a as well as the whole network's recall accuracy. We have also seen a large reduction in error from augmented images with the rotated images having twice the error that flipped images have. It makes sense, therefore, to augment all of the smaller grower datasets.

An additional 1263 augmented images were added to the dataset from growers 87143, 77357, and 32204 and the network retrained (Table 4.14). It appears that the error on grower 7174a has increased beyond its original inaccuracy. We can also see that the rotated and flipped images have errors close to the mean network error (Table 4.15), meaning it is likely dependant on grower 7174a rather than augmented images.

Table 4.14: Comparison of classification error after adding all augmented images

Grower	Original Fail Rate	Augmented 7174a	Augmented 7174a & 98948	Augmented All	Change from 7174a & 98948
Grower - 48785	0.1059	0.0919	0.0939	0.0799	-0.0140
Grower – 27844	0.1112	0.0635	0.0673	0.0943	+0.0270
Grower – 7829e	0.1032	0.0982	0.0861	0.0869	+0.0008
Grower – 5205a	0.0573	0.0362	0.0342	0.0235	-0.0107
Grower – 98948	0.0645	0.0852	0.0496	0.0487	-0.0009
Grower – 9205j	~	~	~	~	~
Grower – 7174a	0.2466	0.2302	0.1737	0.2790	+0.1053
Grower – 87143	0.0764	0.0550	0.1086	0.0567	-0.0519
Grower – 77357	0.0843	0.1306	0.0754	0.1129	+0.0375
Grower – 33204	0.0624	0.0871	0.0847	0.1234	+0.0387
Grower – 26278	~	~	~	~	~
Network Total	0.0995	0.0918	0.0783	0.0918	+0.0135

Table 4.15: Comparison classification errors after adding additional augmented data from all low fruit-count growers to the network

Feature	Classification Failure Rate
Whole Network	0.0918
Rotate Images	0.1156
Flipped Images	0.1046

This error is replicable over multiple training sets. Inspecting the distribution of redness scores, it does not appear that the fruit of Grower 7174a show a tendency to fall near classification boundaries which may have explained its high error rate (Figure 4.9).

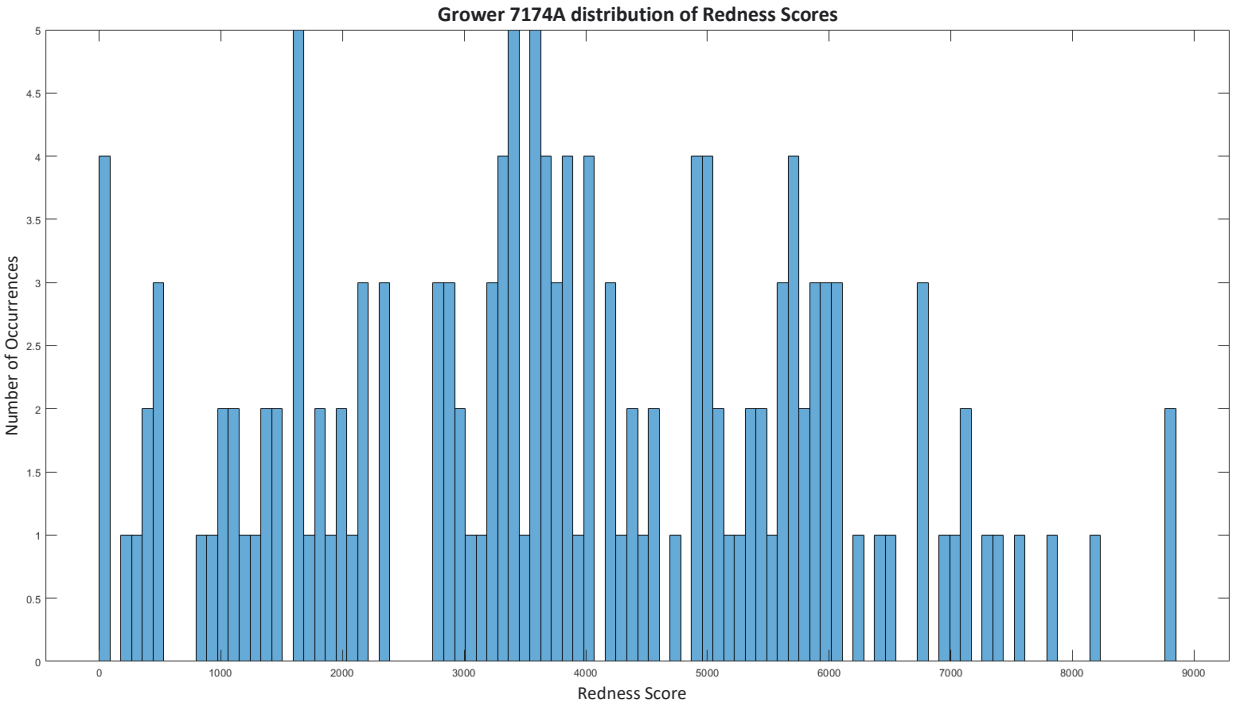


Figure 4.9: Distribution of fruit redness scores for grower 7174a

Because the end goal of this project is to develop a scale of redness that can be used to reliably assess red kiwifruit colour independent of illuminance and spectral sensitivity, the explicit estimation of reflectance allows users to understand how the network reaches its final conclusions on colour quality. This is a very important aspect especially when considering that such a system could determine fruit quality and in turn, grower pay out. With this in mind, exploring how much spectral information is needed to accurately predict redness score is considered imperative in this work.

Considering the errors produced when estimating redness score and kiwifruit class, this may suggest that; (1) kiwifruit of different redness score have the same, or very similar, average spectra. (2) the average spectra of a kiwifruit may not contain enough information to accurately predict a kiwifruit's redness score, and (3) because redness score has a large range (between 0 – 10,000) and classes have been assigned by splitting this range evenly into five classes, scores that fall near the boundary of classes are naturally likely to produce higher errors – this type of error cannot really be avoided.

Finally, memory usage was tracked over the training, validation, and test sets and is shown in table 4.16. The results show that training used less than 1GB of memory and a single inference uses only 0.46GB hence the network is suitable for use on a smartphone.

Table 4.16: Comparison of CNN memory usage

Process	Batch Size	Memory Allocated (Mb)
Training	64	871.75
Validation	64	29.40
Testing	1	0.46

4.3.2 General Regression Neural Network

The next question is how well does the average reflectance of each kiwifruit predict its redness score? And how does compare to estimating score from sRGB images?

Visualising the five classes of kiwifruit (Figure 4.10), it appears that there is significant overlap in spectra between classes. However, considering figure 4.11 and figure 4.12, it appears that between the wavelengths 450nm-700nm there is a pattern emerging between the classes.

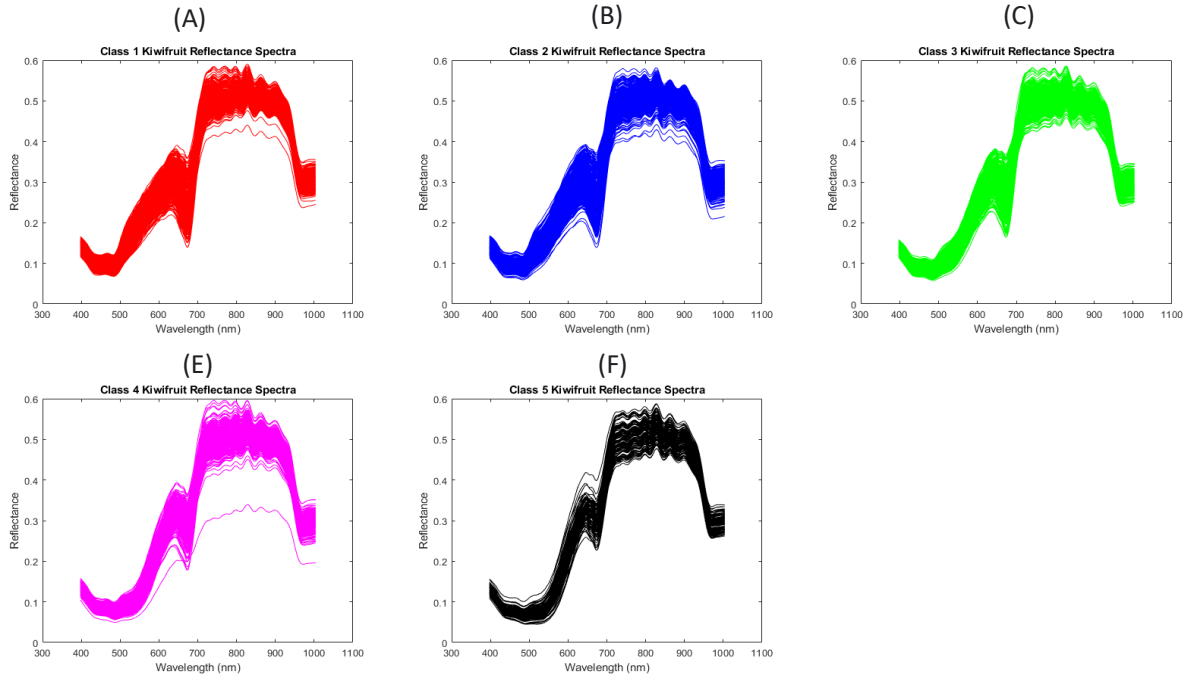


Figure 4.10: Comparison of classes 1(A), 2(B), 3(C), 4(D), and 5 (E) average reflectance spectra.

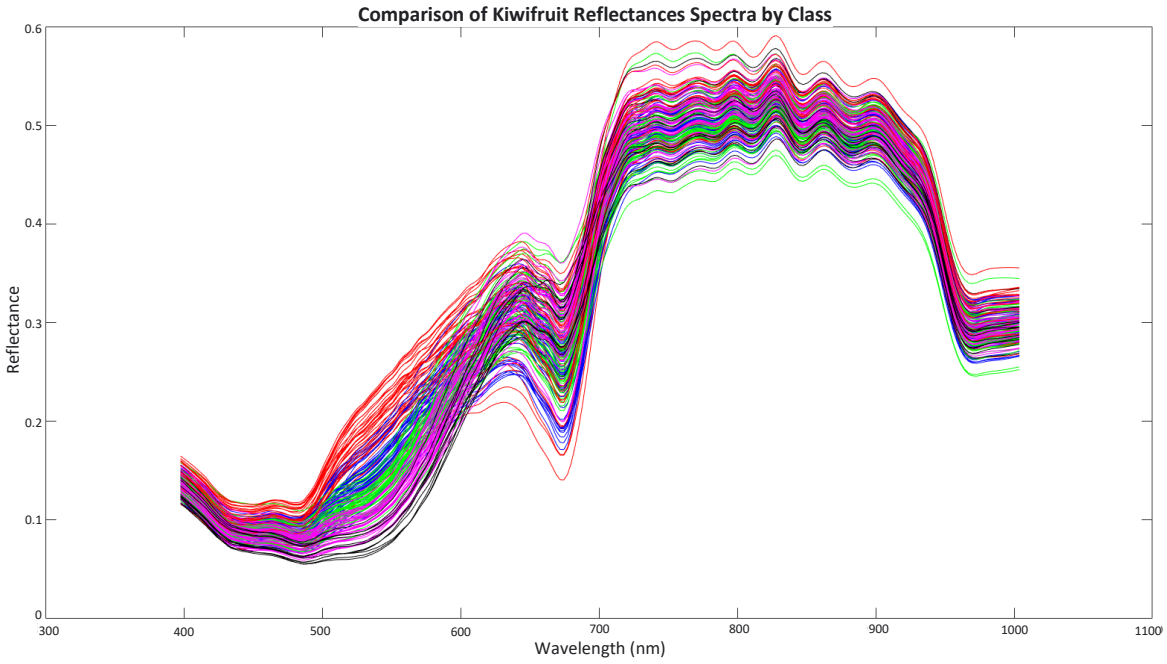


Figure 4.11: Comparison of kiwifruit spectra that fall with 50 redness score of a class boundary where class 1 (red), 2 (blue), 3 (green), 4 (purple), and 5 (black)

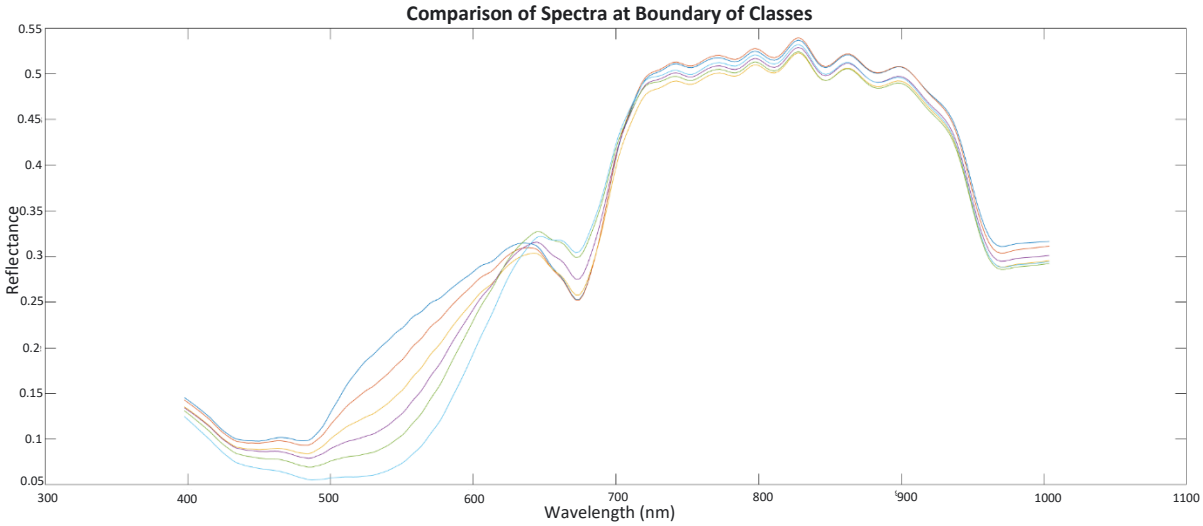


Figure 4.12: Comparison of the average spectra of kiwifruit spectra of redness score 0 (dark blue), 2000 (red), 4000 (orange), 6000 (purple), 8000 (green), 10,000 (light blue)

A typical spectral classification technique SAM was used as a starting point to classify kiwifruit based on the average reflectance of spectra and this produced a classification error of 30% in the visible range and 42% in the visible NIR range. This would suggest that the average reflectance of a kiwifruit may not be sufficient to accurately score it, or that the SAM technique may not be suitable for this application.

In order to increase spectral information, each kiwifruit was segmented into the three key regions, core, inner, and outer pericarp. A sample of a single kiwifruit's segmented spectra is below (Figure 4.13). The outer pericarp appears the most dynamic feature compared to the core and inner pericarp and visually it appears to be the region with the most variation.

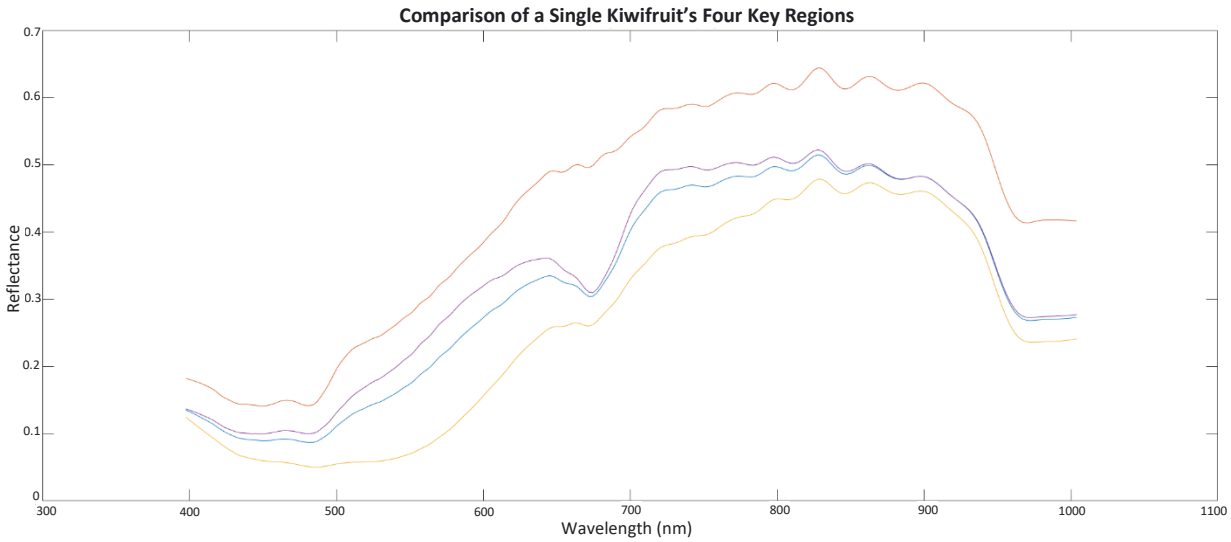


Figure 4.13: Comparison of spectra from the core (orange), outer pericarp (purple), whole fruit (blue), and inner pericarp (yellow)

Initial GRNN Testing

A GRNN was created as per section 3.5. The network was initially trained over 200 epochs using the whole kiwifruit average spectra in the visible range to explore how the loss changed over time (Figure 4.14 and 4.15). Figure 4.14 shows that the validation loss tends to reach a minimum after 75 epochs. Stopping at 100 epochs should be sufficient to assess the network.

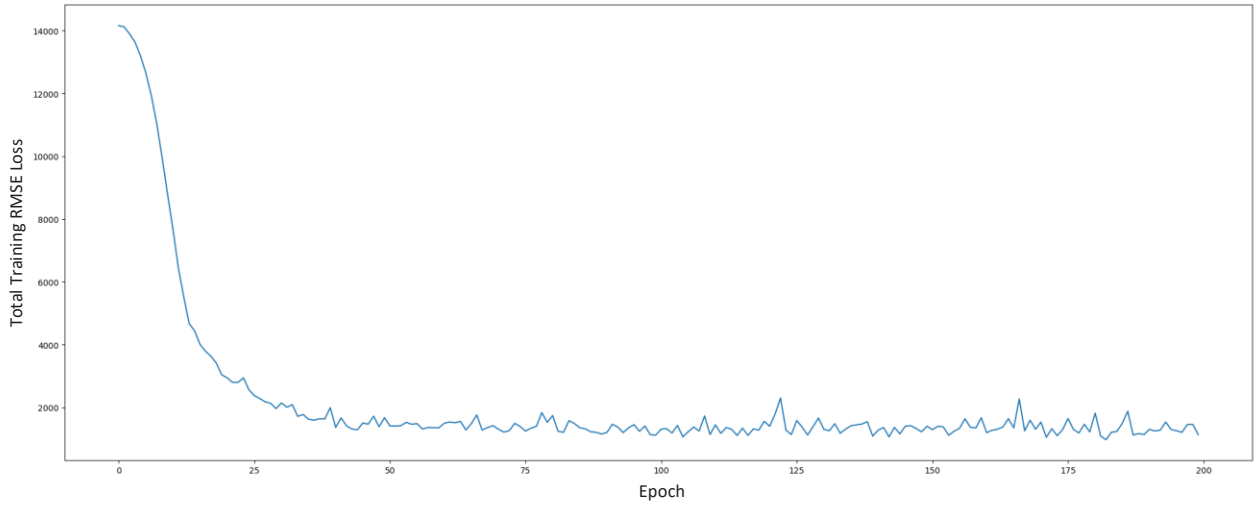


Figure 4.14: Total Training loss of GRNN over 200 epochs

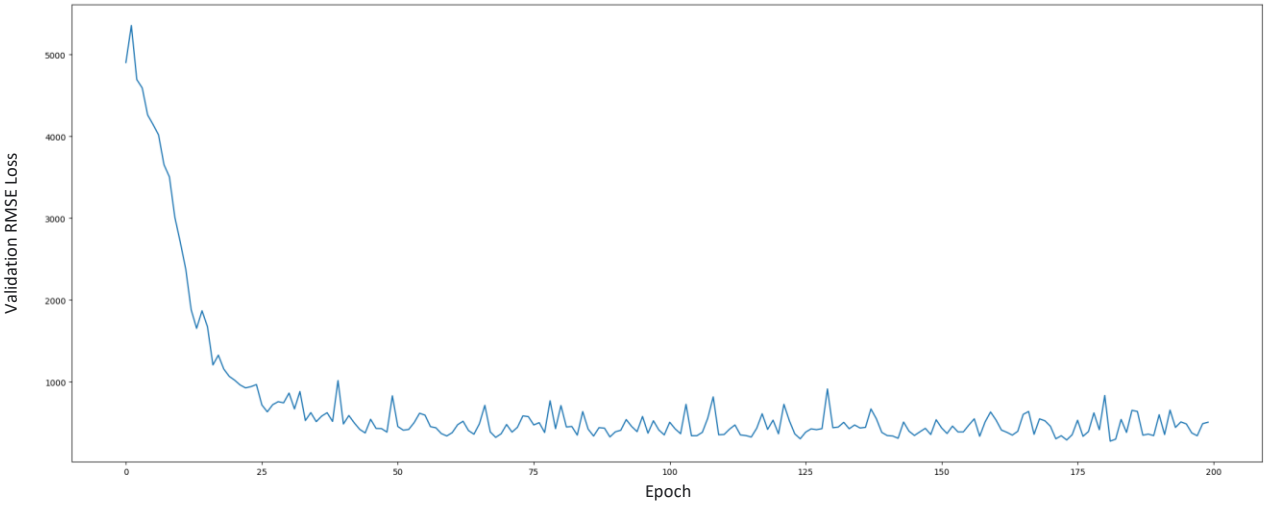


Figure 4.15: Validation RMSE of GRNN over 200 epochs

Next a small experiment to test the size of the GRNN layers was undertaken to explore its effect on validation loss. A GRNN with consistent layer size through the whole network showed better results than using a pattern recognition and summation layer (Table 4.17). Regarding table 4.17, x denotes the size of the input layer and has a value of 143 and 224 for spectra of the visible and NIR spectral ranges respectively. It appears that network 1 or 2 performs better than network 3. The effect of layer size compared to structure 1 and 2 will be investigated further in the project

Table 4.17: Effect of layer size on network error where x=143 for the visible range, x=224 for the NIR range

Network Name	Network Structure	Test RMSE	Mean Error	Med Error	Std Error	Effc Fail Rate
1	$L1 = x$ $L2 = x$ $L3 = x$ $L4 = 1$	268.16	207.63	151.25	173.75	0.0960
2	$L1 = x$ $L2 = 2x$ $L3 = x$ $L4 = 1$	227.31	182.60	159.98	138.20	0.1017
3	$L1 = x$ $L2 = 2x$ $L3 = x/2$ $L4 = 1$	358.90	251.56	194.23	284.37	0.1356

Estimating Kiwifruit Redness Score from Average Reflectance

The effect of each key kiwifruit segment was investigated by feeding the average reflectance of each segment into the GRNN. The results (Figures 4.16 to 4.22) show that in the visible range both the whole average and outer average segments produce the lowest errors. As expected, the core does not appear to explain redness score. The inner pericarp appears to contain some information although not nearly as much as the whole and outer segments.

The results suggest that so long as the whole or outer segment's average reflectance spectra is provided as an input to the network in either the visible or visible NIR range, the network will produce a mean error of around 200 with a standard deviation of around 150. A table of results can be found in appendix 8.1.

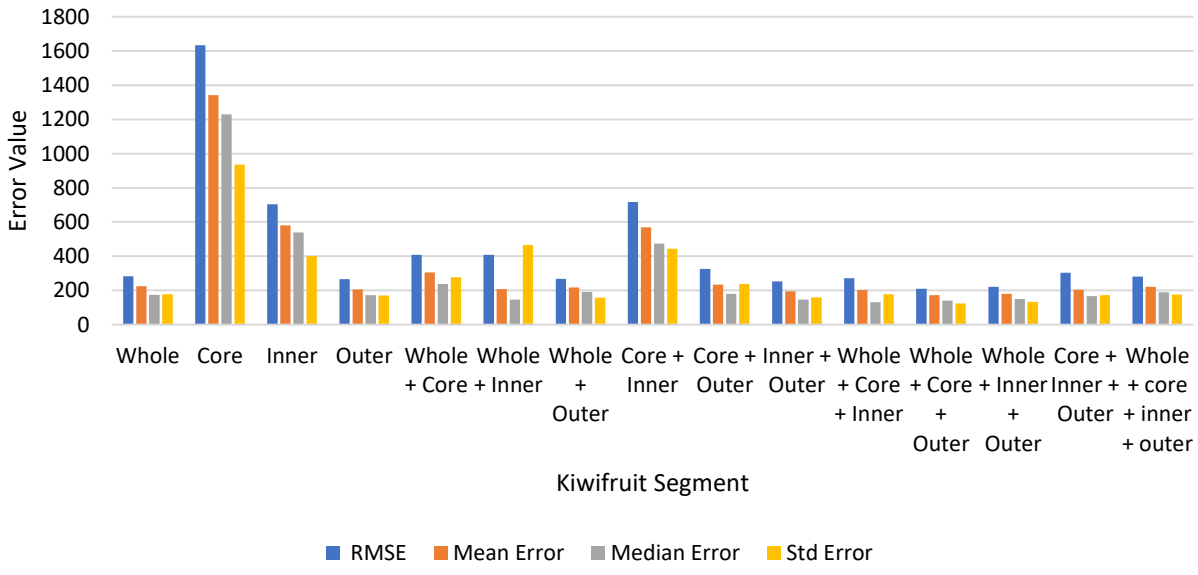


Figure 4.16: Effect of kiwifruit segment average reflectance spectra on different error metrics in the visible range

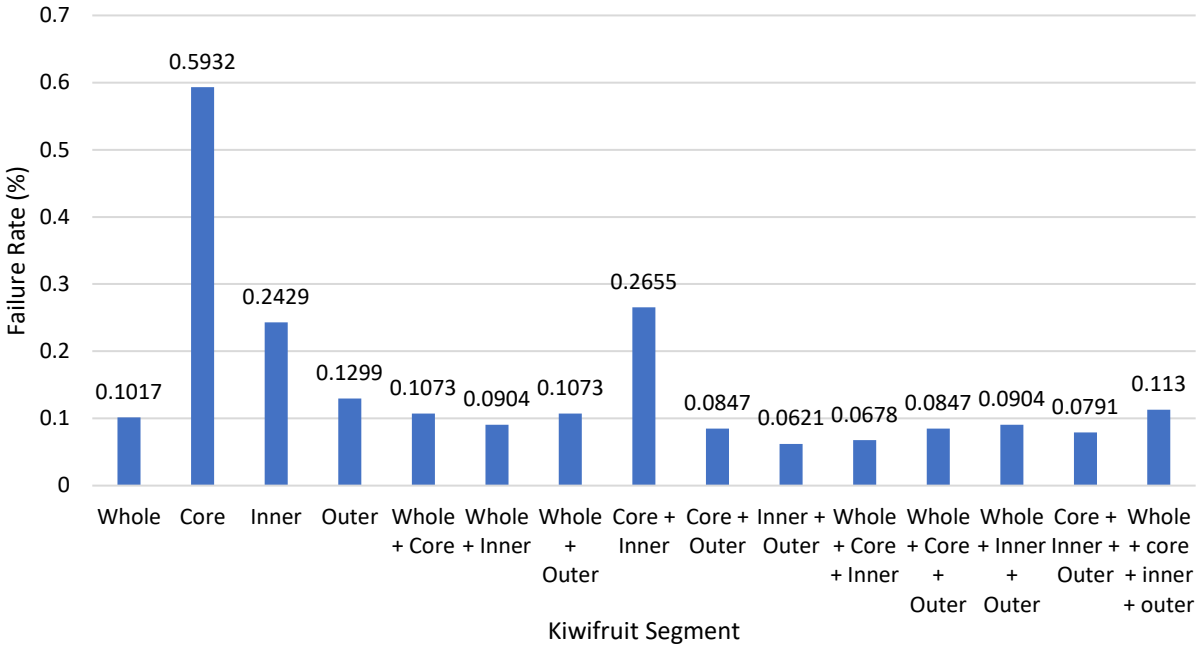


Figure 4.17: Effective Failure Rate of different kiwifruit segments as network Inputs over the visible range

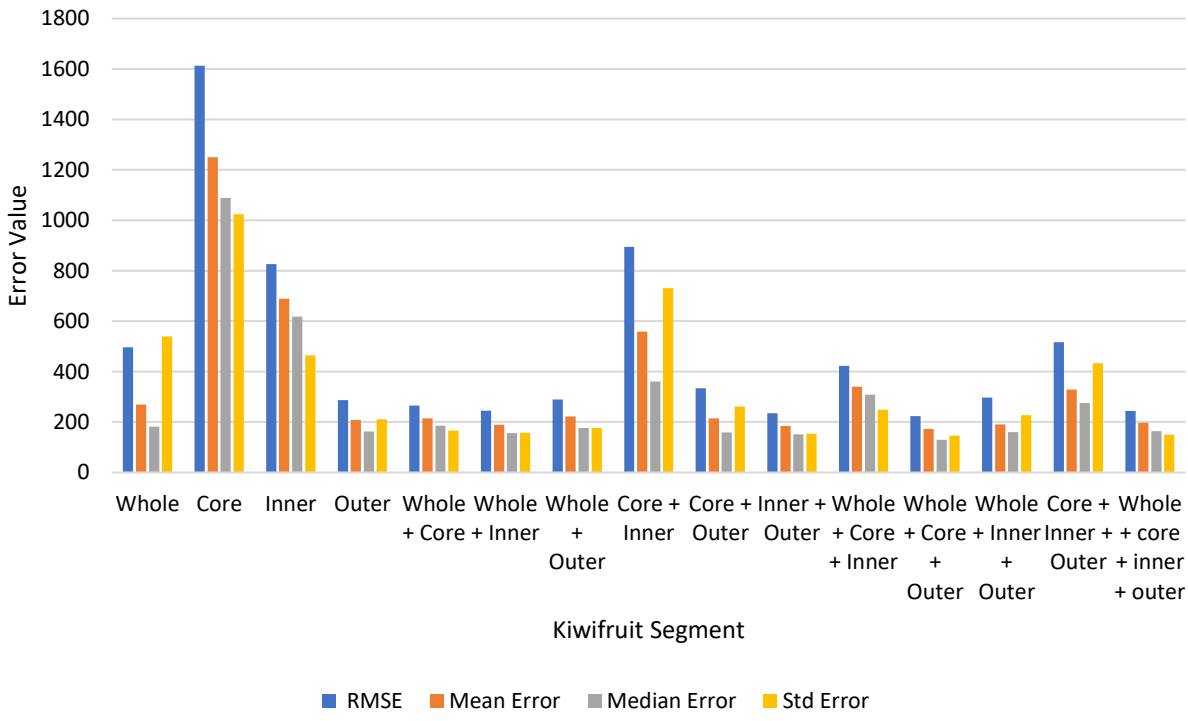


Figure 4.18: The Effect Of Kiwifruit Segment Average Reflectance Spectra On Different Error Metrics In The Visible Range

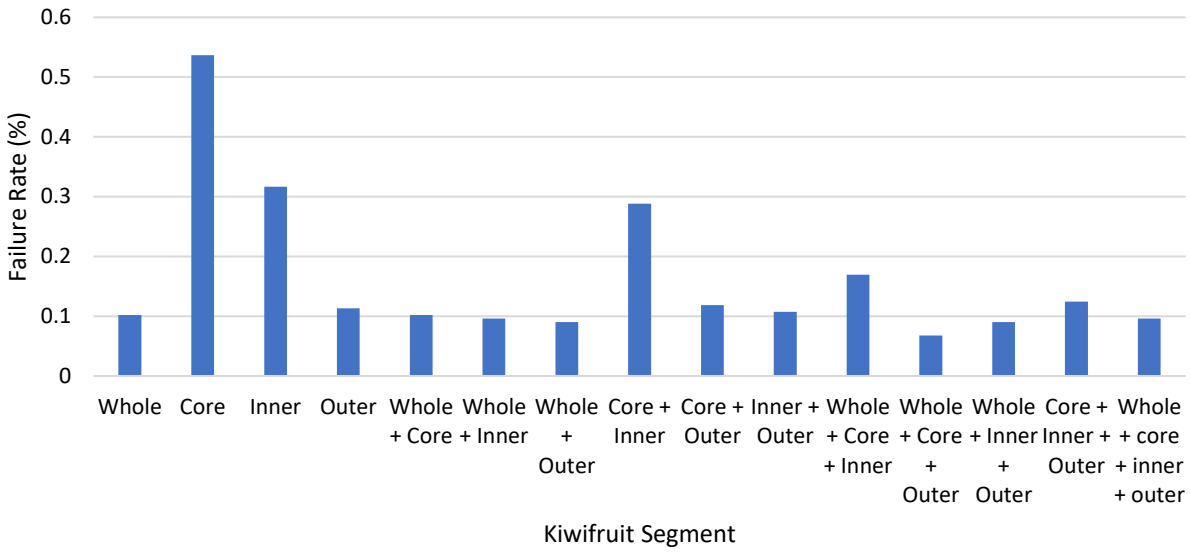


Figure 4.19: Effective Failure Rate of different network inputs in the visible NIR range

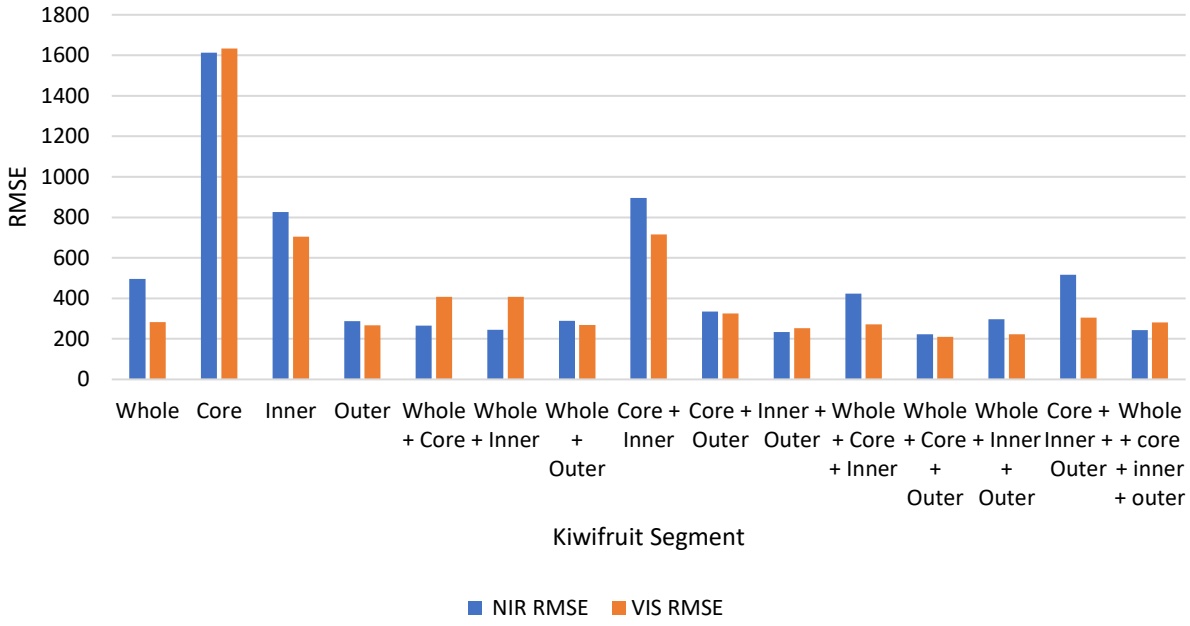


Figure 4.20: Comparison Of The Effect Of Using VIS Or VIS+NIR Spectra As Network Inputs On Network RMSE

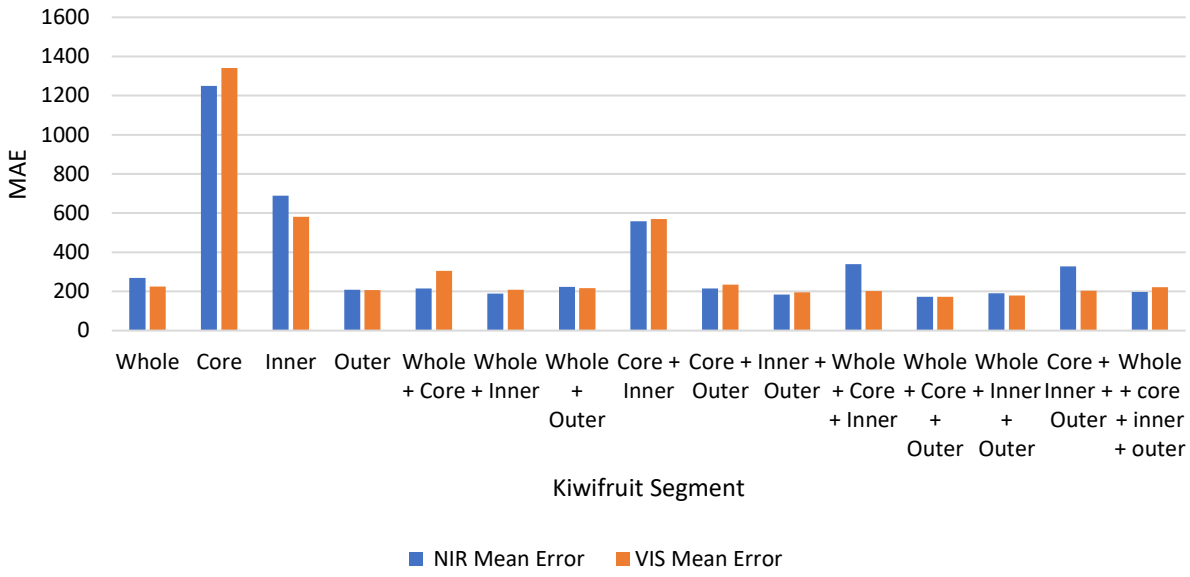


Figure 4.21: Comparison Of The Effect Of Using VIS Or VIS+NIR Spectra As Network Inputs On Network Mean Error

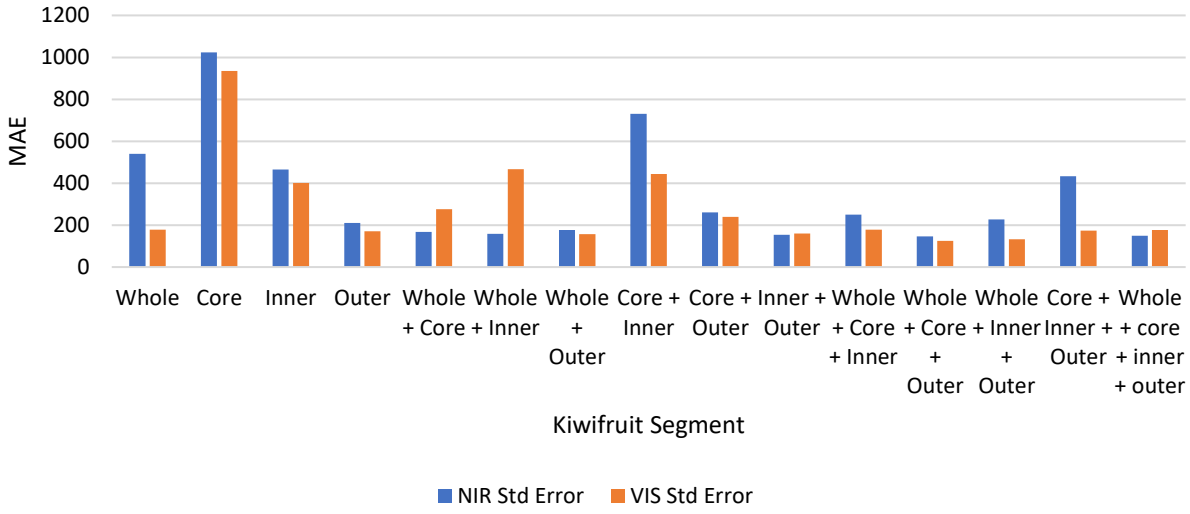


Figure 4.22: Comparison of the effect of using VIS or VIS+NIR spectra as network inputs on network Mean Error

Finally, table 4.18 outlines the memory used to trained and test the network. The results show that training and validation use minimal memory and are suitable for use on a smartphone.

Table 4.18: Memory usage by GRNN

Process	Batch Size	Memory Allocated (Kb)
Training	64	191.5
Validation	64	36
Testing	1	1

4.3.3 Hyperparameter Tuning of GRNN

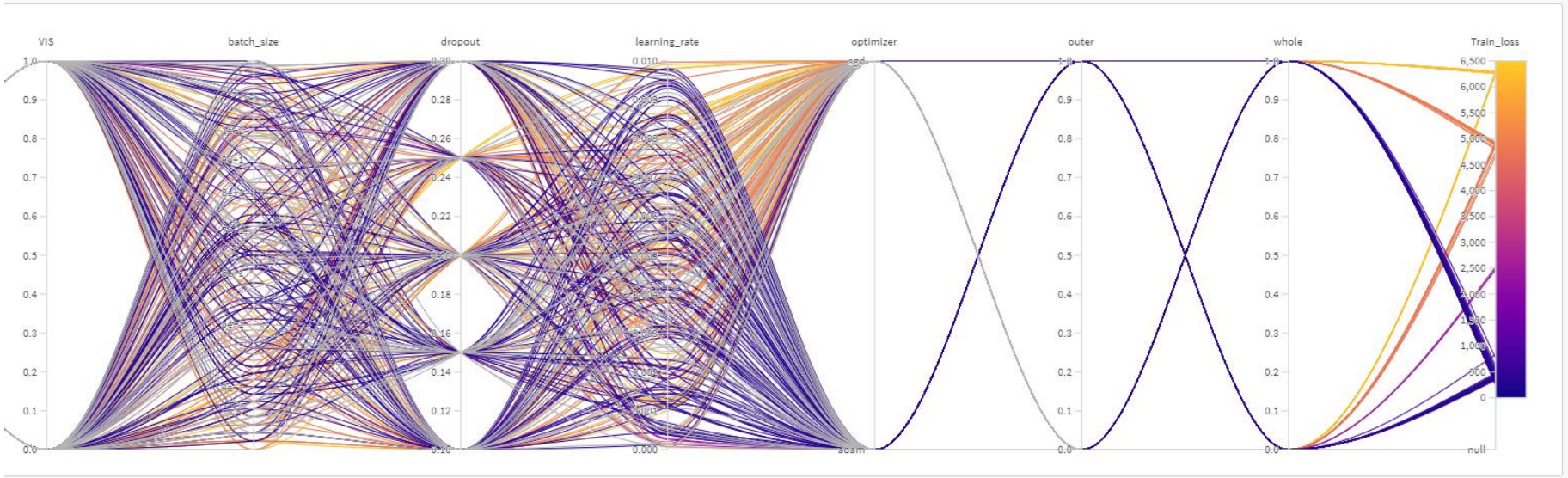
An optimisation of GRNN hyperparameters was undertaken using wandb package. An initial parameter sweep was created using the configuration found in appendix 8.4.1.

A total of 300 sweeps were completed. The results show that the Adam optimiser vastly outperforms the SGD optimiser (Figure 4.23). Over the full 300 sweeps, the results suggest that using the outer segment *and* whole segment were negatively correlated with the error over the test set. This may be noise, or it may suggest that using the outer *and* whole kiwifruit segments decreases network error compared to using just one or the another. Similarly, learning rate was negatively correlated with test error which suggests a higher learning rate performs better for this GRNN.

Comparatively, the visible range, dropout percentage, and batch size are all positively correlated with error over the test set which suggests using the visible NIR spectral range, a smaller dropout layer percentage, and lower dropout percentage will decrease the network's error moving forward.

The sweep graphs (Figure 4.23) show the values of each parameter for a single training set. Each line on a sweep graph relates to a single training run. The line intersects different parameters such as learning rate and batch size, before showing the final RMSE value achieved on the test set.

(A)



(B)

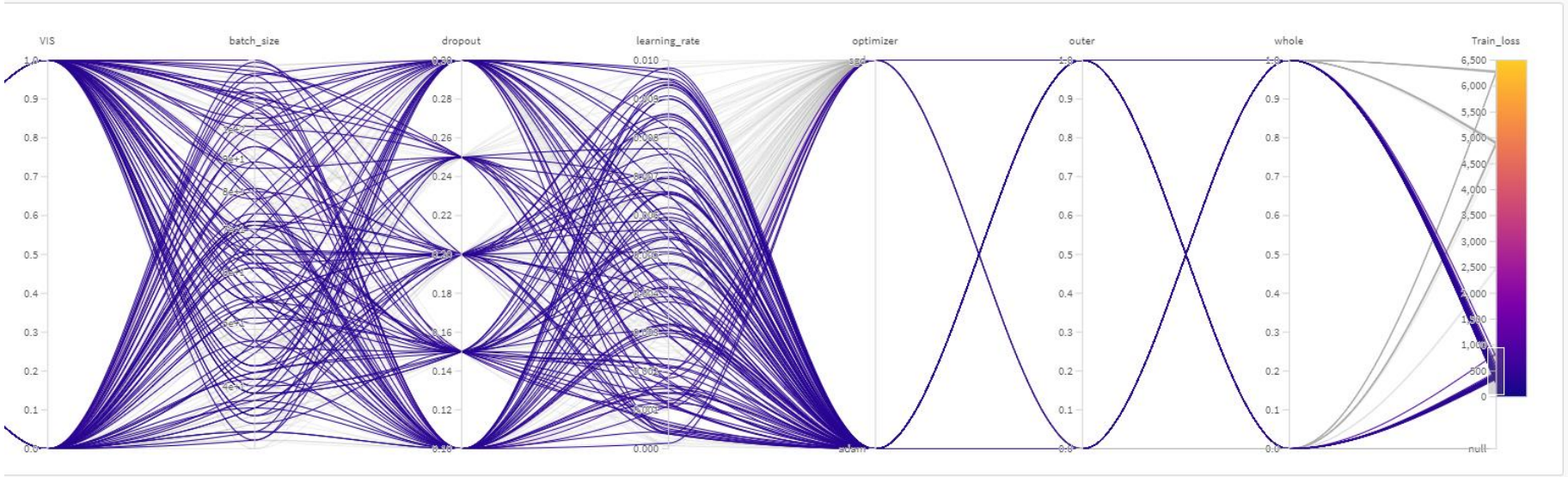


Figure 4.23: Sweep graph showing all 300 sweeps (A) and best performing sweeps (B)

A second sweep was created with the configuration shown in appendix 8.4.2. This sweep explores the effect of layer size related to the second and third layer of the network. This sweep varied each layer between 0.5 and 1.5 of its original size compared to the network inputs.

The results of this sweep mostly agree with the previous sweep regarding learning rate, dropout percentage, and visible vs visible NIR ranges. Interestingly, the results show that the first layer size is weakly positively correlated to the test set error whereas the second layer size is weakly negatively correlated. This would suggest that a smaller second layer and a larger third layer may decrease network error.

A final sweep was then created that extended the range of the multiplier to the second and third layer to be between 0.25 and 2 of the input layer size. A total of 757 sweeps were completed and this showed that both the size of layers 2 and 3 are weakly negatively correlated to the network's error which suggests layers that are larger than the input layer provide smaller network errors.

4.3.4 Spectral Invariance to Illumination

In section 4.3.1, we used only the RAW images captured directly from the Samsung Galaxy S8 and iPhone 12 smartphones. The results showed that a single convolutional layer with an additional 3 dense layers provided predictions of a kiwifruits average reflectance spectrum across different spectral sensitivities and camera angles with an RMSE of 0.0113.

To explore the model's ability to learn invariance to a scene's illumination, a set of synthetic data was created using equation 2.9. The database provided by (Tominaga, Nishi, & Ohtera, 2021) contains 20 different smartphones' spectral sensitivities. Whilst having such a wide array of data to choose from is ideal, it is impractical to use every smartphone sensitivity to generate data as some phones share the same sensor and more so, using all the data available would produce a dataset of $20 * 5 * 1857 = 185{,}700$ datapoints and would significantly outweigh the real RAW data.

To begin, 10 smartphones were selected from the available 20 to ensure the dataset included common smartphones such as iPhone and Samsung devices and that any phones with repeat sensors were not included (Table 4.19). An initial dataset of 5000 images was created. Additional data will be generated depending on the output from initial testing.

Table 4.19: Smartphones and sensor types

Smartphone	Sensor (Tominaga, Nishi, & Ohtera, 2021)
iPhone8	Sony IMX315
iPhone11	Sony IMX503
iPhone12 Pro Max	Sony IMX603
Samsung Galaxy Note 10 Pro	Samsung ISOCELL S5K2L4
Samsung Galaxy S20	Samsung ISOCELL S5KGW2
Huawei P10 Lite	Sony IMX214
Sharp Aquos R5G	Infineon Technologies IRS2381C
Xiaomi Redmi Note 9S	Samsung ISOCELL S5KGM2
Sony Xperia 5-II	Sony IMX557
Google Pixel 4	Sony IMX363

A comparison of synthetic and real image error metrics on the validation set is shown in table 4.20. There has been a significant improvement in the RMSE values of the synthetic data compared to the real data (Table 4.20) and the results show that synthetic data produces a lower RMSE value compared to real data and this difference is approximately three standard deviations different.

Table 4.20: Comparison of errors when training and testing on real vs synthetic data

Spectral Range	MAE	MSE	RMSE	Std of RMSE
Synthetic - VIS	0.1834	0.000046	0.0068	0.0032
Synthetic - VIS-NIR	0.1187	0.000043	0.0065	0.0024
Real -VIS	0.2024	0.0002	0.0131	0.0044
Real -VIS-NIR	0.2144	0.0001	0.0113	0.0049

The results also show (Figure 4.25) that the integral error is lowest for synthetic data in the visible range, although considering the other error metrics, the NIR range appears better regarding MRAE and SAM scores (Figure 4.26).

The performance metrics by datatype, phone type, grower, and fruit size across the visible and NIR ranges can be found in figures 4.27 through 4.29. There does not appear to be any one attribute that causes significantly higher or lower errors compared to the network average. This suggests that training on a combined dataset will not produce higher error than compared to using on real images.

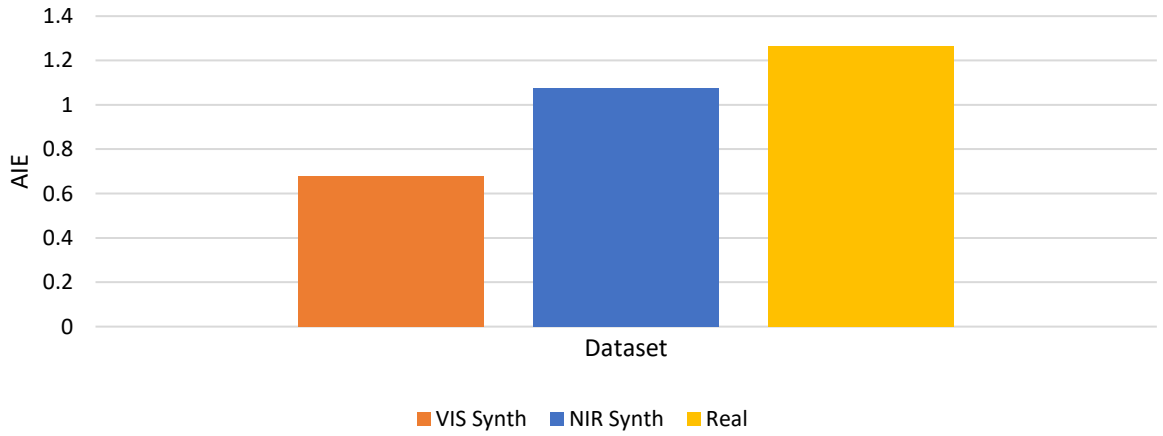


Figure 4.24: Comparison of integral error of synthetic data

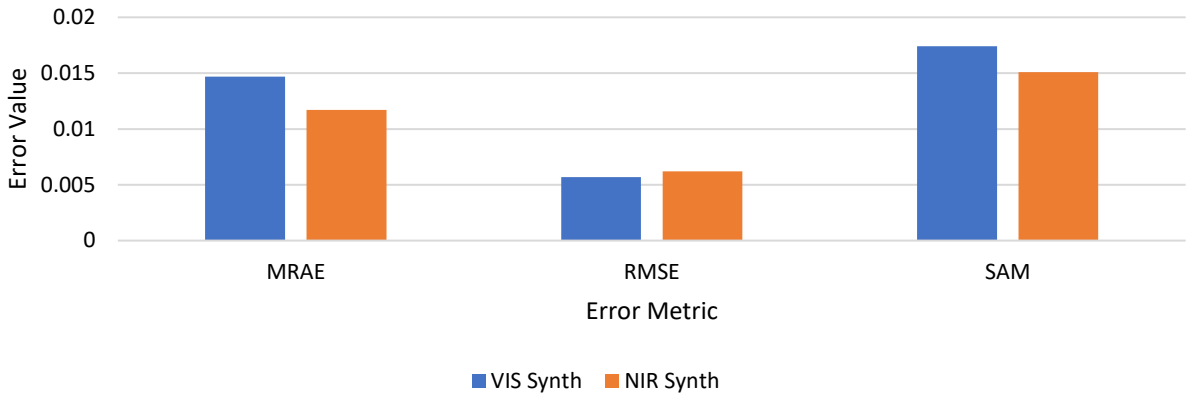


Figure 4.25: Comparison of errors in synthetic data of the visible and visible NIR spectral ranges

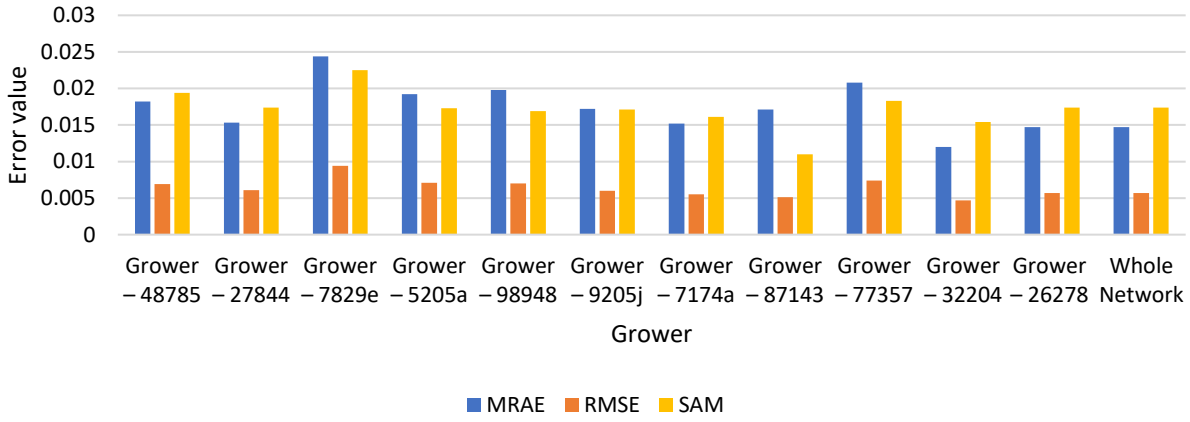


Figure 4.26: Comparison of Error Metric by Grower Over the Visible Range

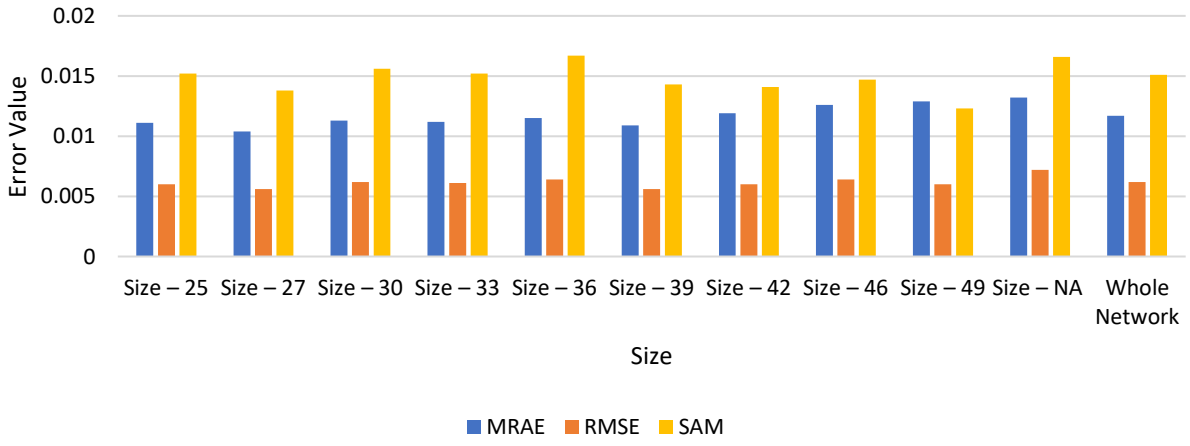


Figure 4.27: Comparison of Error Metric by Fruit Size Over the Visible Range

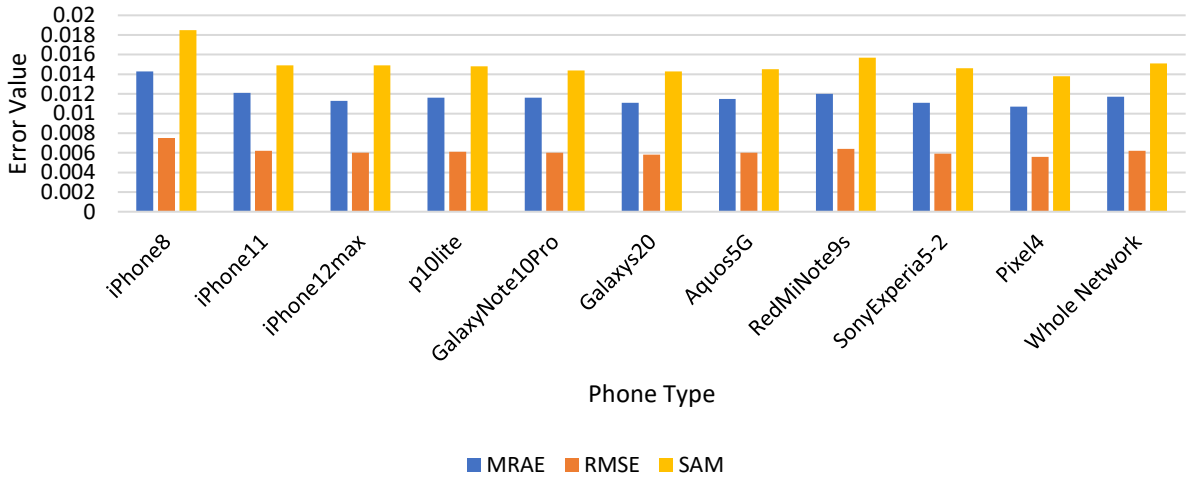


Figure 4.28: Comparison of Error Metric by Phone Type Over the Visible Range

A full set of results can be found in Appendix 8.2.

Next, the RAW images from the iphone12 and the Samsung Galaxy s8 were combined to create a dataset of both real and synthetic images. The dataset totalled 11,927 images and the network was retrained (50 epochs) (Table 4.21). Whilst our model error metrics are comparable across all datasets, it appears that when using the combined dataset there is a clear difference in error metrics between the raw and synthetic data (Figure 4.30, 4.31). The error of the combined dataset falls, as expected, between the RMSE values of the real and synthetic dataset.

Table 4.21: Comparison network errors from real, synthetic, and combined datasets

Spectral Range	MAE	MSE	RMSE
Combined - VIS	0.1502	0.000121	0.0109
Combined - VIS-NIR	0.1302	0.000092	0.0096
Synthetic - VIS	0.1834	0.000046	0.0068
Synthetic - VIS-NIR	0.1187	0.000043	0.0065
Real -VIS	0.2024	0.000211	0.0145
Real -VIS-NIR	0.2144	0.000135	0.0116

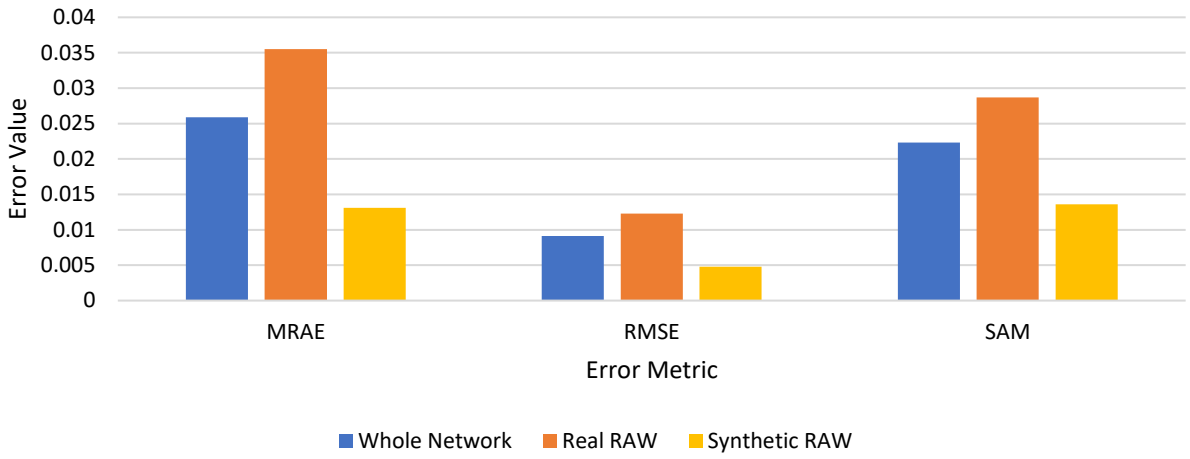


Figure 4.29: Comparison of Error Metrics of Real vs Syntenic data from Combined Dataset Over the Visible Range

The error metrics by grower, fruit size, phone type, and data type can be found in figures 4.31 through 4.37. There appears to be a slightly high error on size 46 in the visible range (Figure 4.32) and the real phone images also produce higher than average errors across both the visible and NIR range (Figure 4.33 and 4.36).

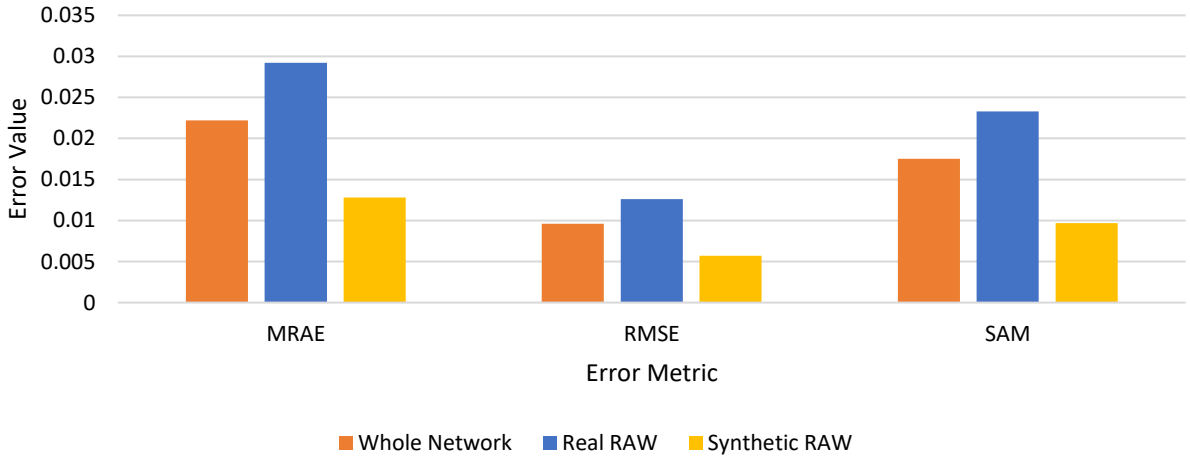


Figure 4.30: Comparison of Error Metrics of Real vs Synthetic data from Combined Dataset Over the Visible NIR Range

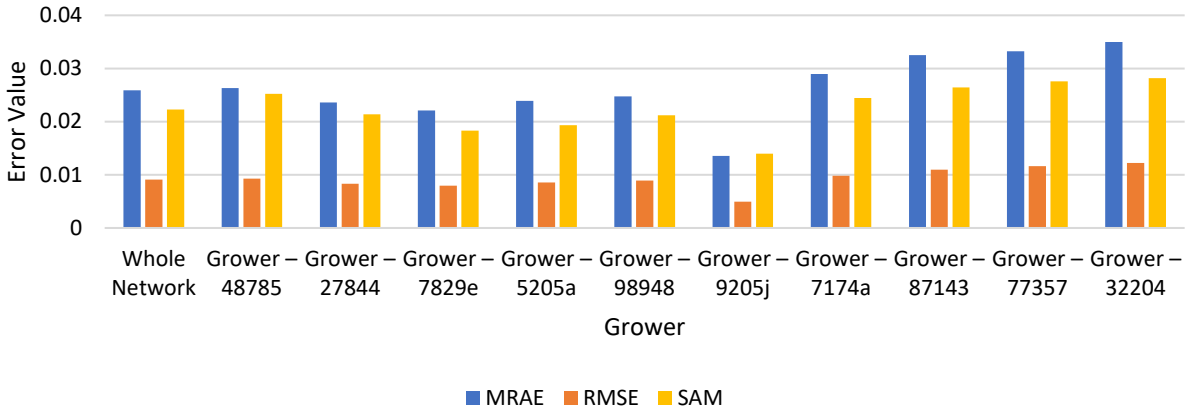


Figure 4.31: Comparison of Error Metrics by Grower Using the Combined Dataset over the visible range

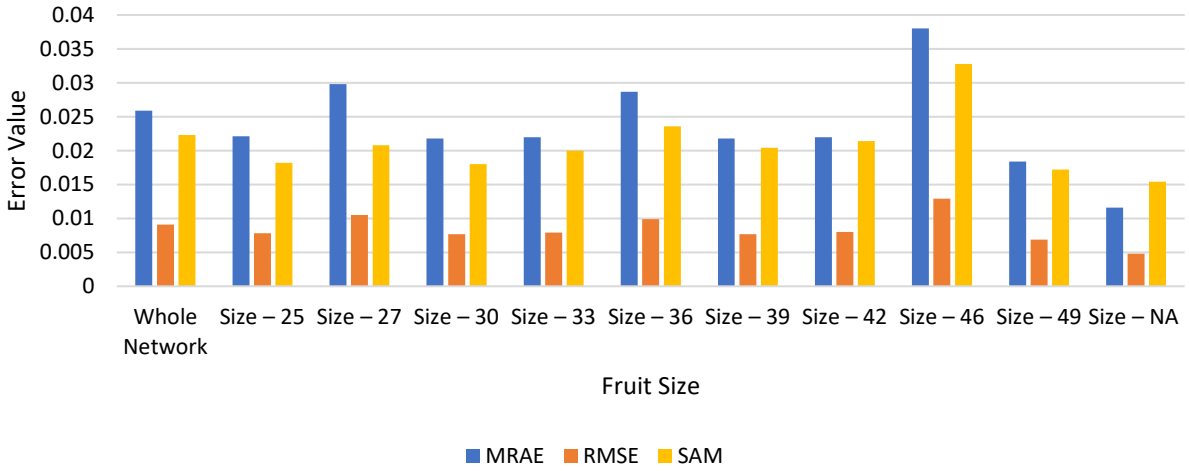


Figure 4.32: Comparison of Error Metrics by Fruit Size Using the Combined Dataset over the visible range

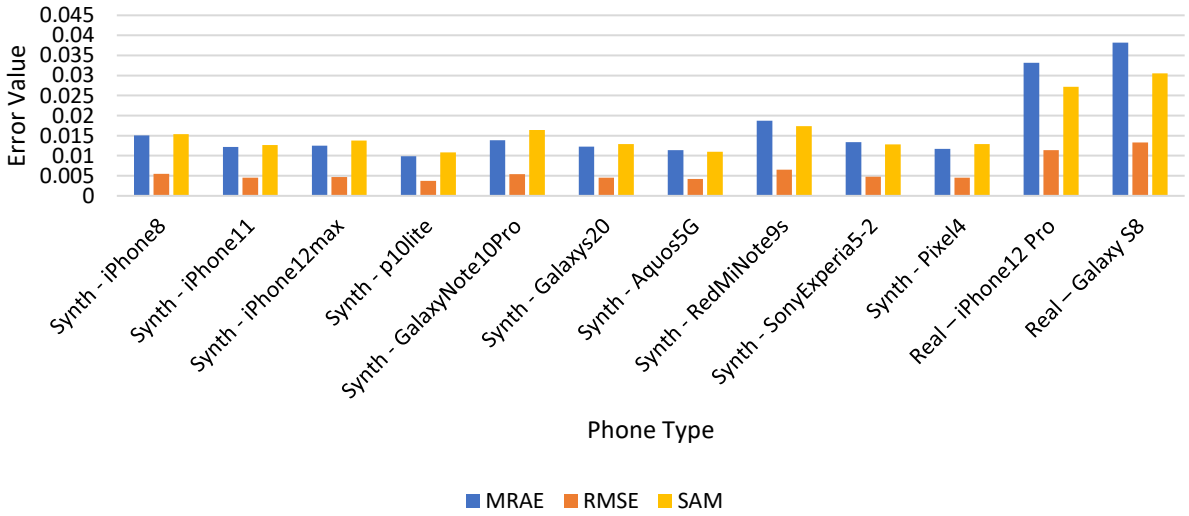


Figure 4.33: Comparison of Error Metrics by Phone Type Using the Combined Dataset over the visible range

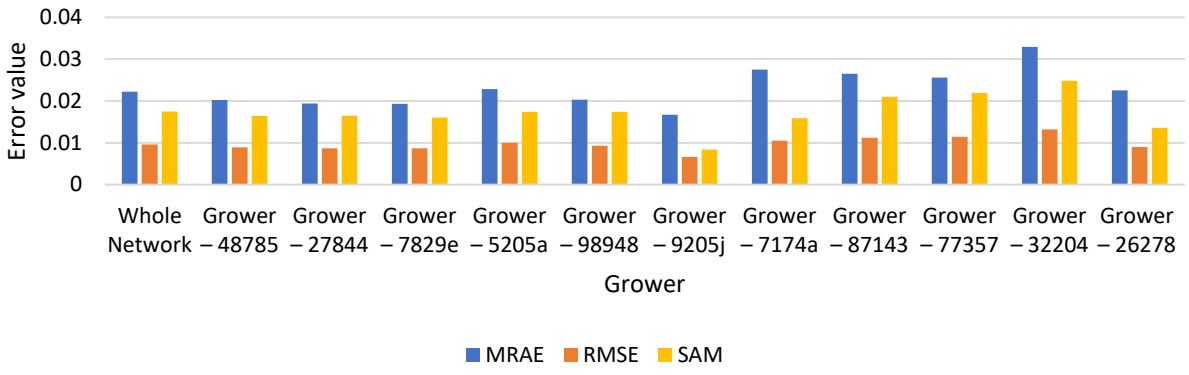


Figure 4.34: Comparison of Error Metrics by Grower Using The Combined Dataset over the visible NIR range

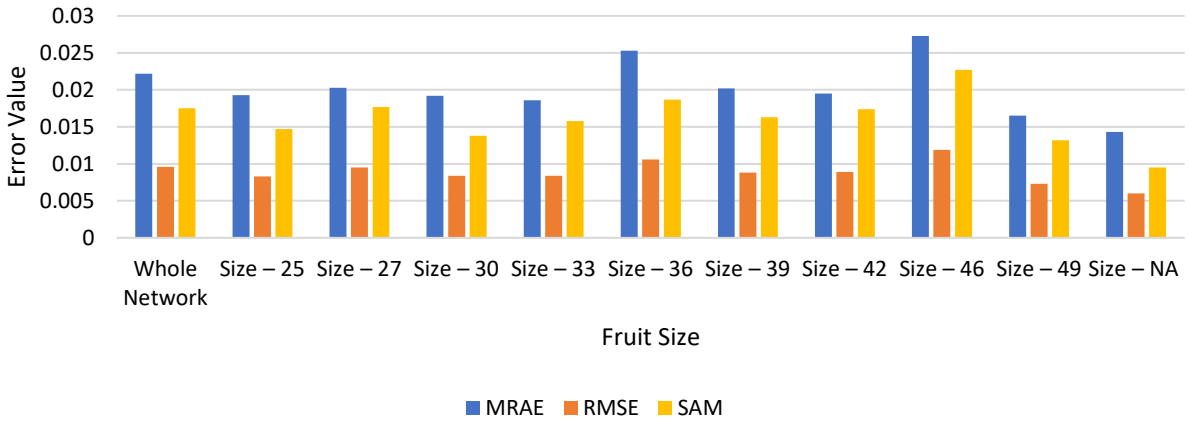


Figure 4.35: Comparison Of Error Metrics By Fruit Size Using The Combined Dataset Over The Visible NIR Range

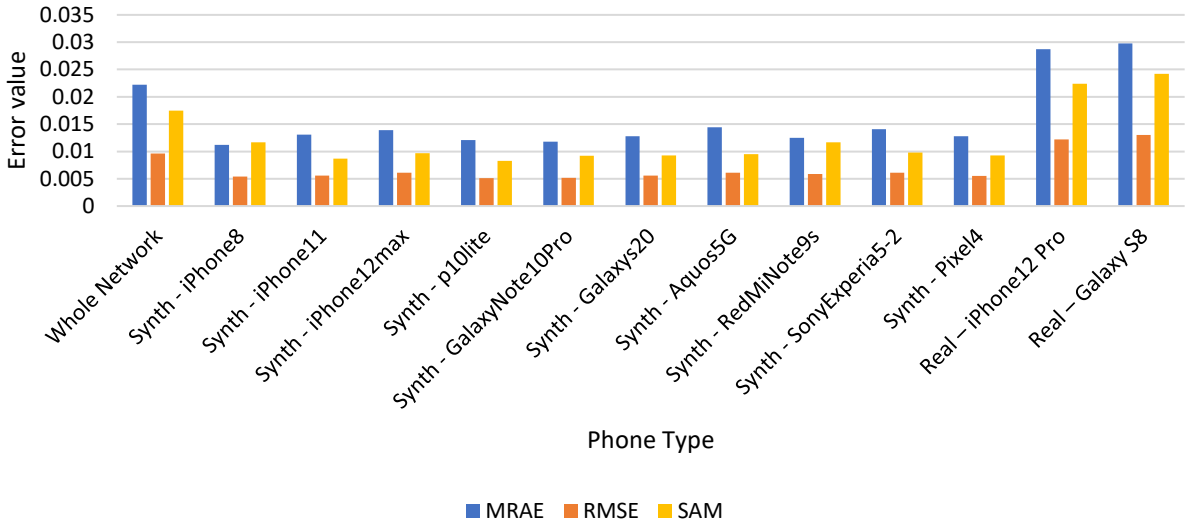


Figure 4.36: Comparison Of Error Metrics By Phone Type Using The Combined Dataset Over The Visible NIR Range

A full set of results can be found in Appendix 8.3. To check that the network has sufficient epochs to reach a minimal error, the validation RMSE was plotted against the number of training epochs. As can be seen in figure 4.37, the network reaches a minimum value around ~15 epochs

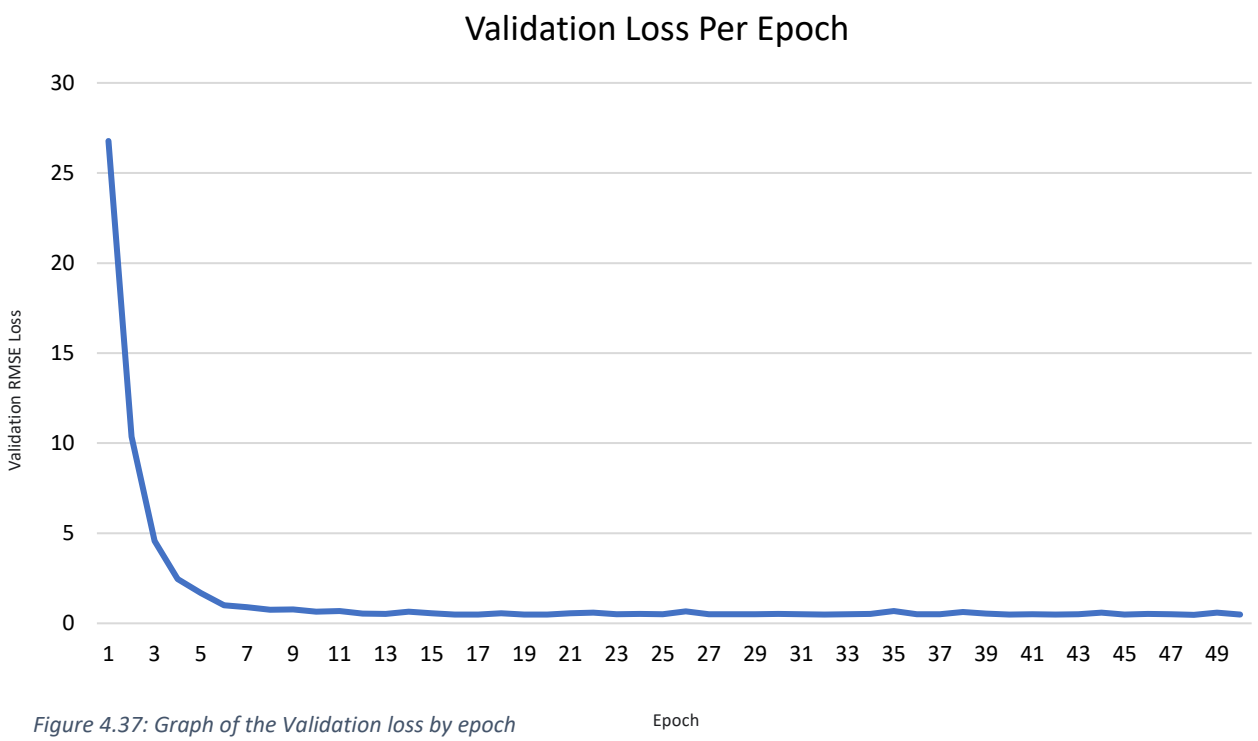


Figure 4.37: Graph of the Validation loss by epoch

To further explore synthetic and real-world data, a test was carried out using synthetic data generated from the iPhone 12 Pro Max spectral sensitivity (Tominaga, Nishi, & Ohtera, 2021) and the iPhone 12 Pro RAW images captured during this project. A reliable source for the exact sensor that is used in the iPhone 12 Pro could not be found. However, these are the closest models of phones that we have both RAW and synthetic data of.

This experiment sought to understand the validity of using synthetic data in place of real data by training the network on the synthetic data and validating it using the real data and vice versa.

It was noticed that fruit synthesised from hyperspectral data appear 90° rotated from those captured with smartphones (Figure 4.37). To control for this rotation the real dataset was rotated 90° clockwise so that the angles in both datasets matched (Figure 4.39). After rotation, the images were then laterally inverted so they aligned completely (Figure 4.40).

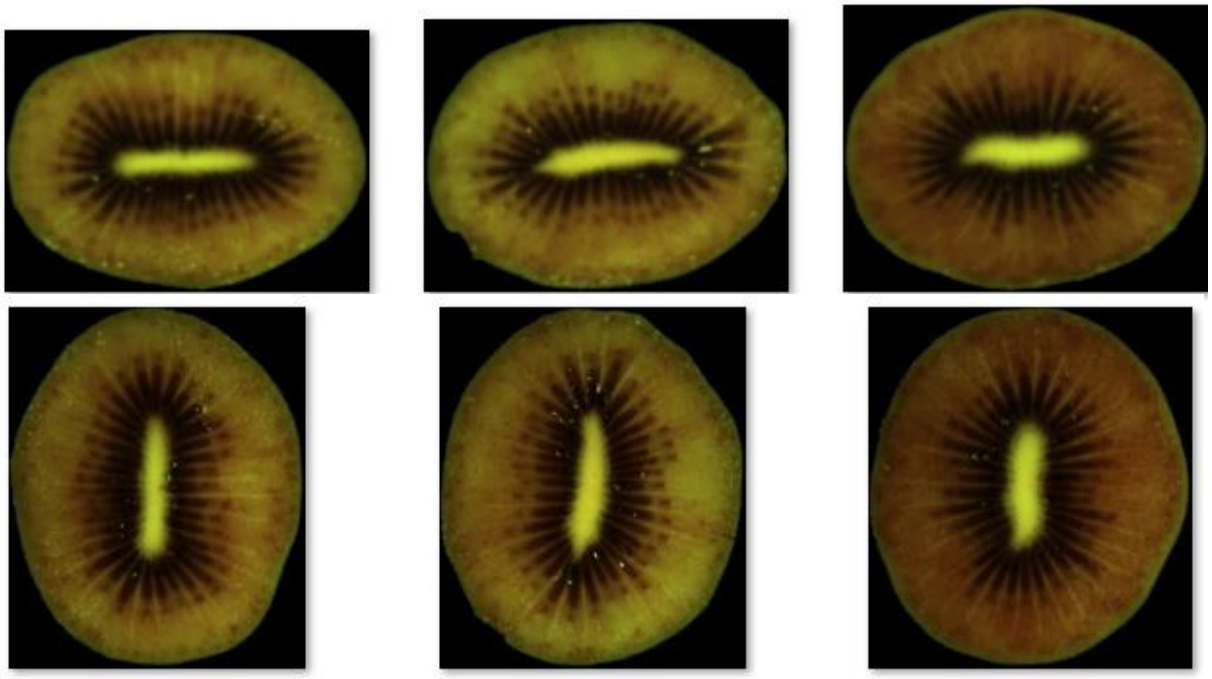


Figure 4.38: Comparison of Synthetic (top) and real (bottom) RAW images before rotation

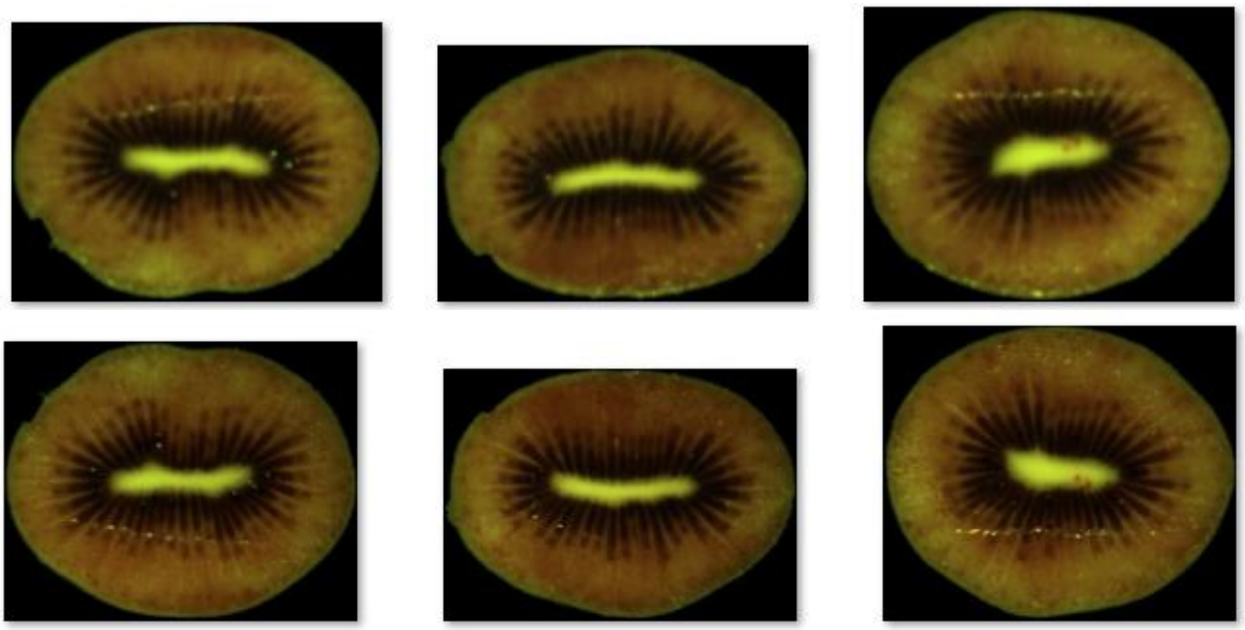


Figure 4.389: Comparison of Synthetic (top) and real (bottom) RAW images after rotation

The synthetic images appear slightly blurred in comparison to the real images in these figures as the synthetic images are generated from the hyperspectral data which has a low spatial resolution of around 250x250px. However, when input into the network, all images are resized to 200x200px.

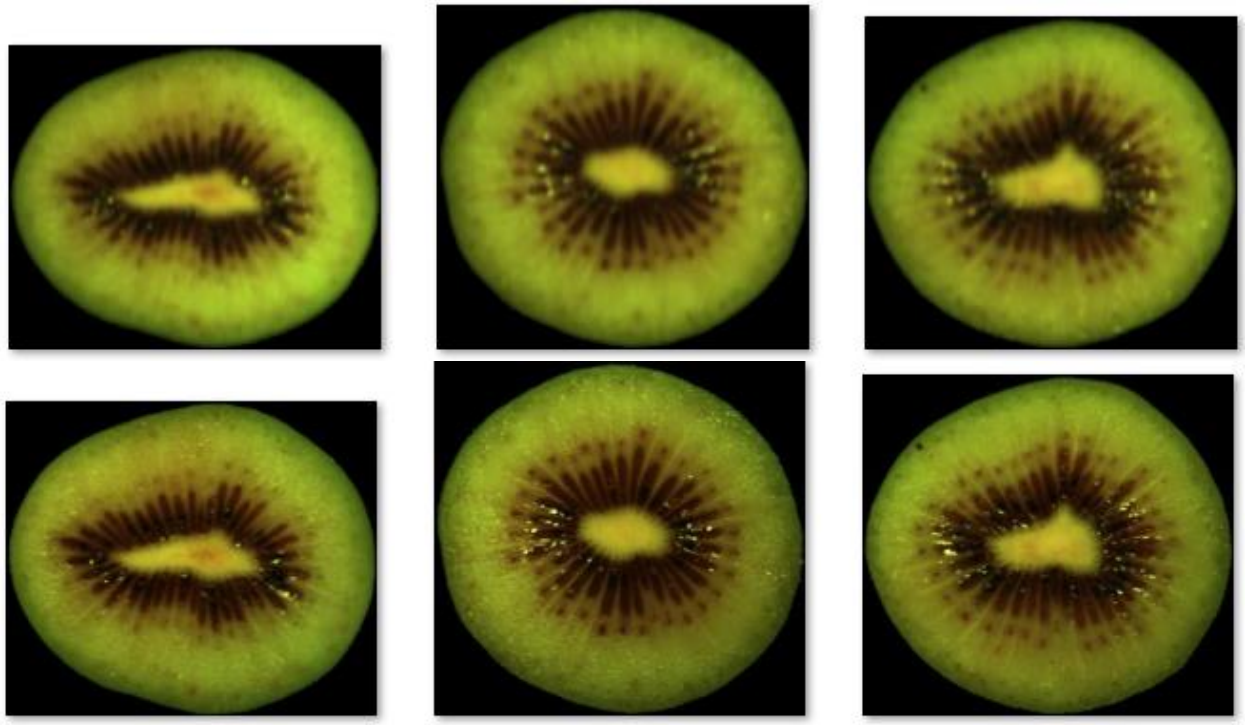


Figure 4.39: Comparison of Synthetic (top) and real (bottom) RAW images after lateral inversion

It appears that training the network on synthetic vs real and validating it on opposing datasets produces similar results (Table 4.22).

Table 4.22: Comparison of network error using real or synthetic data for training and validation

Metrics NIR Range	MAE	MSE	RMSE
Training (Synthetic)	0.1748	0.0000825	0.00886
Validation (Real)	0.2383	0.0001556	0.01242
Training (Real)	0.1472	0.0000572	0.00739
Validation (Synthetic)	0.2608	0.0001830	0.01337

It appears that both networks show similar results (Table 4.23) and that training a network with synthetic data, in the case of kiwifruit, has been successful. This means that it is best to use the combined dataset to maximise spectral and illuminant variance in further training of the model. As this experiment controlled for the orientation of kiwifruit, it is likely that the high errors produced previously (Figure 4.33 and 4.36) when using the combined datasets can be attributed to the orientation of kiwifruit. This can be addressed by adding randomly rotating images so the network can learn invariance to rotation.

Table 4.23: Comparison of errors using real or synthetic data for training and validation

NIR RANGE	Count	AIE	MRAE	RMSE	SAM
Train (synth) Val (Real)	1704	2.1458	0.0279	0.0124	0.0226
Train (Real) Val (Synth)	1733	2.3508	0.0262	0.0134	0.0304

4.3.5 KiwiNet

The previous sections of this chapter have validated the use of a CNN and GRNN to estimate the average reflectance of a red kiwifruit and how to corelate this reflectance to a final redness score. To conclude the study, the architectures of the CNN and GRNN were combined to create a single network. This network has been named Kiwinet (Figure 4.41).

Initially the model was initialised as one large network (figure 4.41) and the loss was calculated from the error between the estimated and true redness score (Table 4.24). It can be seen from the output of the CNN part of the network that the estimated reflectance values are far from the true reflectance values (Table 4.25).

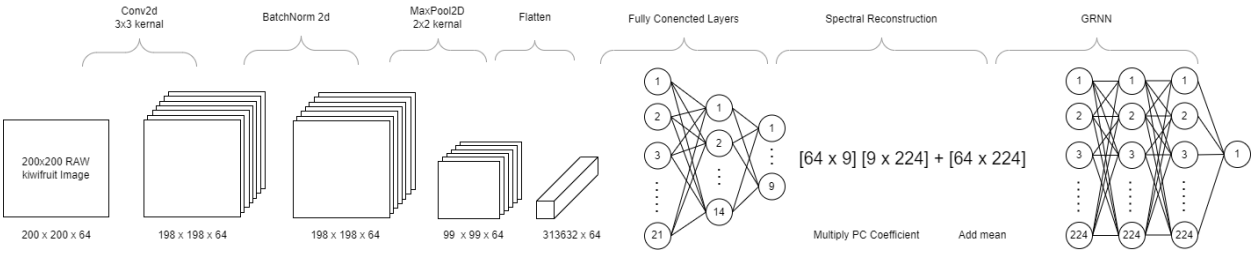


Figure 4.40: KiwiNet Architecture for batch size of 64 when predicting 9 principal components (NIR). A 200x200px demosaiced image is input into the network where it is convolved using a 3x3px kernel. The result is normalised and max pooled using a 2x2px kernel. This result is then flattened and passed through 3 fully connected which each predict a single principal component of the reflectance spectra. The spectrum is then fully reconstructed and passed through the 3 fully-connected layers of the GRNN to produce a final redness score.

Table 4.24: Redness score error from Kiwinet

	Count	Mean Error	Median Error	Standard Deviation
Redness Score	2908	384.55	288.43	339.69

Table 4.25: Reflectance errors from Kiwinet

	Count	AIE	MRAE	RMSE	SAM
Reflectance	2908	8254.49	59.84	67.35	1.34

When visualising the reflectance errors from the network (Figure 4.42), it appears that the network is estimating an abstracted version of the kiwifruit's reflectance (which will now being referred to as pseudo-reflectance) but only over the range 400-650nm. Particularly, a single point value at approximately 590nm could be measured to represent redness score. The inflection point of the pseudo-reflectance is at the wavelength of 583nm. This assumes that the network is plotting psudo-reflectance in wavelength space, which cannot be assumed. The wavelength space will be referred to as pseudo-wavelength.

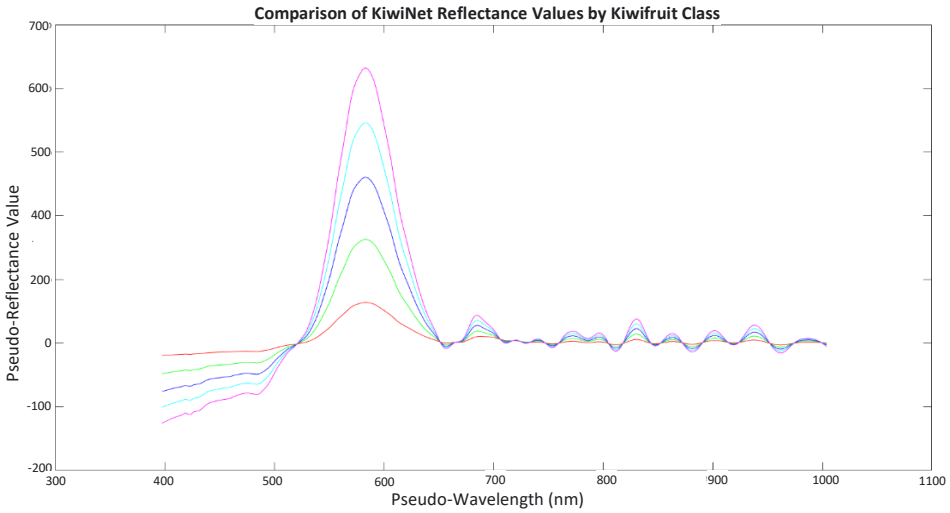


Figure 4.41: Pseudo-reflectance output from KiwiNet plotted against wavelength by kiwifruit class where red=class 0, green=class 1, purple=class 2, blue=class 3, pink=class 5 (class 0 is very green and class 4 is very red in appearance)

Figure 4.43 shows the appearance of kiwifruit at 583nm. There appears to be a clear pattern that as redness score increases the fruit appear darker at this wavelength. 583nm represents signals that appear yellow to the eye and considering yellow can also be achieved through the mixing of red and green primaries, it is likely that this single point reading is another way of representing the subtraction of red and green RGB channels of a kiwifruit image. Figure 4.44 shows a clear trend between the value of the pseudo-reflectance and kiwifruit class

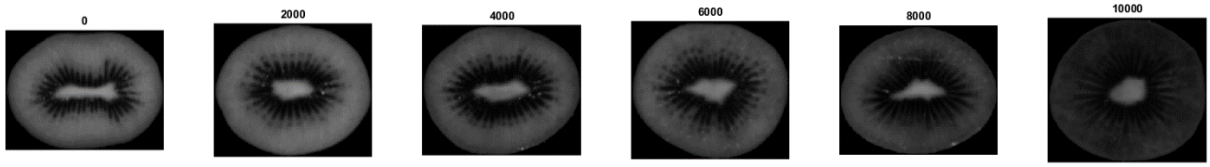


Figure 4.42: Sample of a range of kiwifruit with different redness score at 583nm

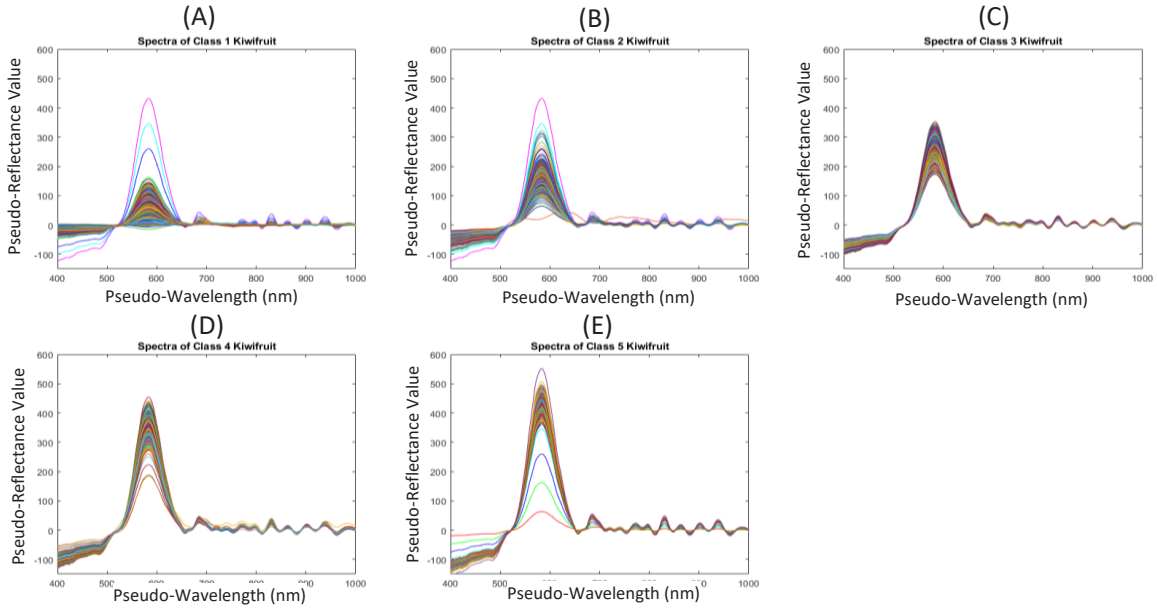


Figure 4.43: Pseudo-reflectance from KiwiNet by fruit class

In place of the larger conjoined network, the CNN and GRNN were trained separately and then the output from the validation set of the 50th epoch was passed into the GRNN to evaluate the overall performance of the whole system (Table 4.26).

Table 4.26: Redness Score errors from KiwiNet using real reflectance

	Count	Mean Error	Median Error	Standard Deviation
Kiwinet Redness	2908	220.33	171.00	213.82

It appears that the combination of the CNN and GRNN has provided redness score results with similar errors to those from the ground truth reflectance data. This would suggest that the GRNN network has some robustness to spectral errors that are produced by the CNN as the final RMSE match those from the ground truth data. This error rate would lead to a correct classification rate of 91.06%. A histogram depicting the distribution of redness score errors can be found in figure 4.45.

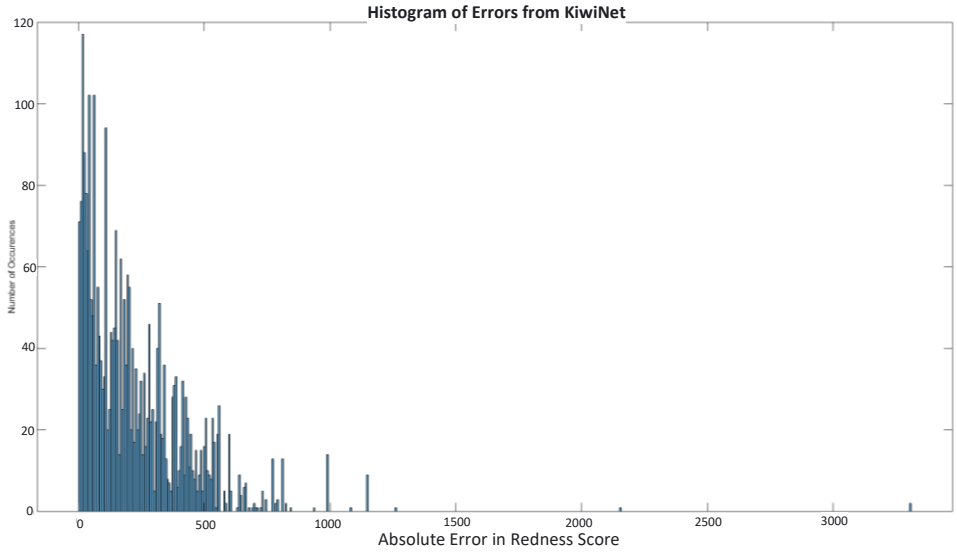


Figure 4.44: Histogram of absolute errors in redness score from KiwiNet.

Lastly, a proof-of-concept mobile application was created to ensure the CNN and GRNN models could be deployed on a typical smartphone. An app was created using Android Studio that is compatible with Android versions Lollipop and higher. The app was tested on a Samsung Galaxy S8 device. The app allows a user to take an image or select one from their gallery and then pass the image through the CNN and GRNN models separately (Figure 4.47).

The app demonstrates that both the CNN and GRNN can be deployed on a typical smartphone. Figure 4.47 shows the reflectance output from an AdobeRGB image as, at the time of writing, we do not have any Red19 kiwifruit to take images of. Further testing can be carried out over the next harvest season. Hence the pseudo-reflectance shown in figure 4.46 does not align with the reflectance presented in previous sections of this thesis as the image in figure 4.46 is in AdobeRGB format whilst the CNN model was trained on demosaiced RAW images. This pseudo-reflectance does, however, show a distinct dip in its spectrum that

aligns with those of the reflectances trained on raw images. This suggests that the model has learned some invariance to image datatypes.

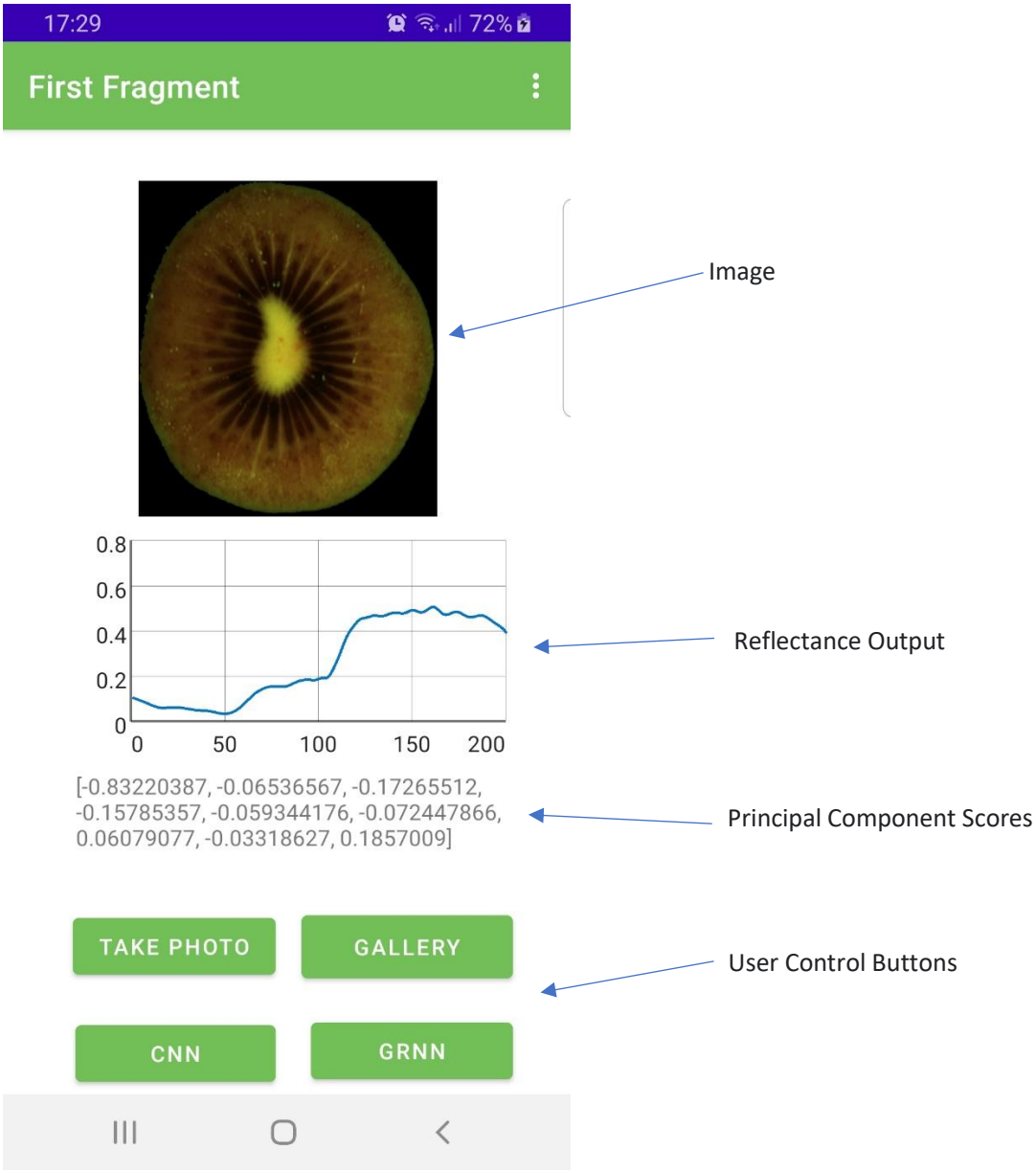


Figure 4.45: Figure of KiwiNet mobile application showing the estimated pseudo-reflectance of a single kiwifruit.

4.3.6 Network Visualisation

The filters of the CNN trained as part of KiwiNet and trained independently can be found in Appendix 8.3. It appears that the filters do not match and considering KiwiNet is outputting pseudo-reflectance as opposed to actual reflectance, it is expected that these filters would look different. There does not appear to be any obvious patterns in either set of filters.

The output of a single kiwifruit from each filter from KiwiNet (Figure 4.47) and when the CNN is trained separately (Figure 4.48) are shown below. There does not appear to be any obvious trends or similarities between them.

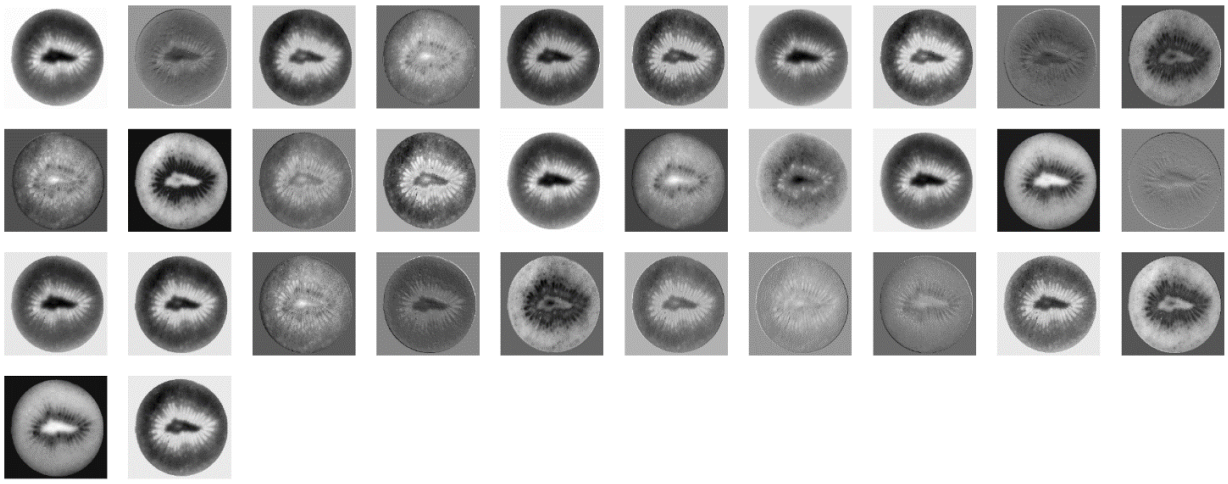


Figure 4.46: Visualisation of single kiwifruit after convolution from 32 filters of KiwiNet when trained together.

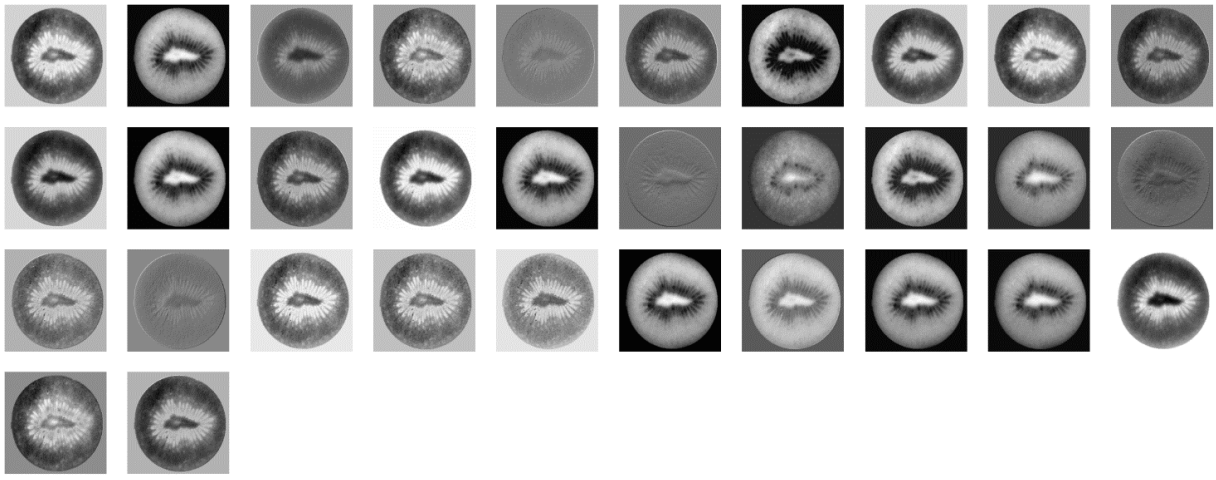


Figure 4.47: Visualisation of single kiwifruit after convolution from 32 filters of KiwiNet when trained separately.

During testing, additional convolutional layers were added to see if they improved reflectance estimation. Whilst they did not significantly improve the error metrics, when visualising the fifth convolutional layer from this testing, there appears to be only two different images output from 64 filters.

One of the two images appears to represent the kiwifruit image at 583nm closely (Figure 4.50)

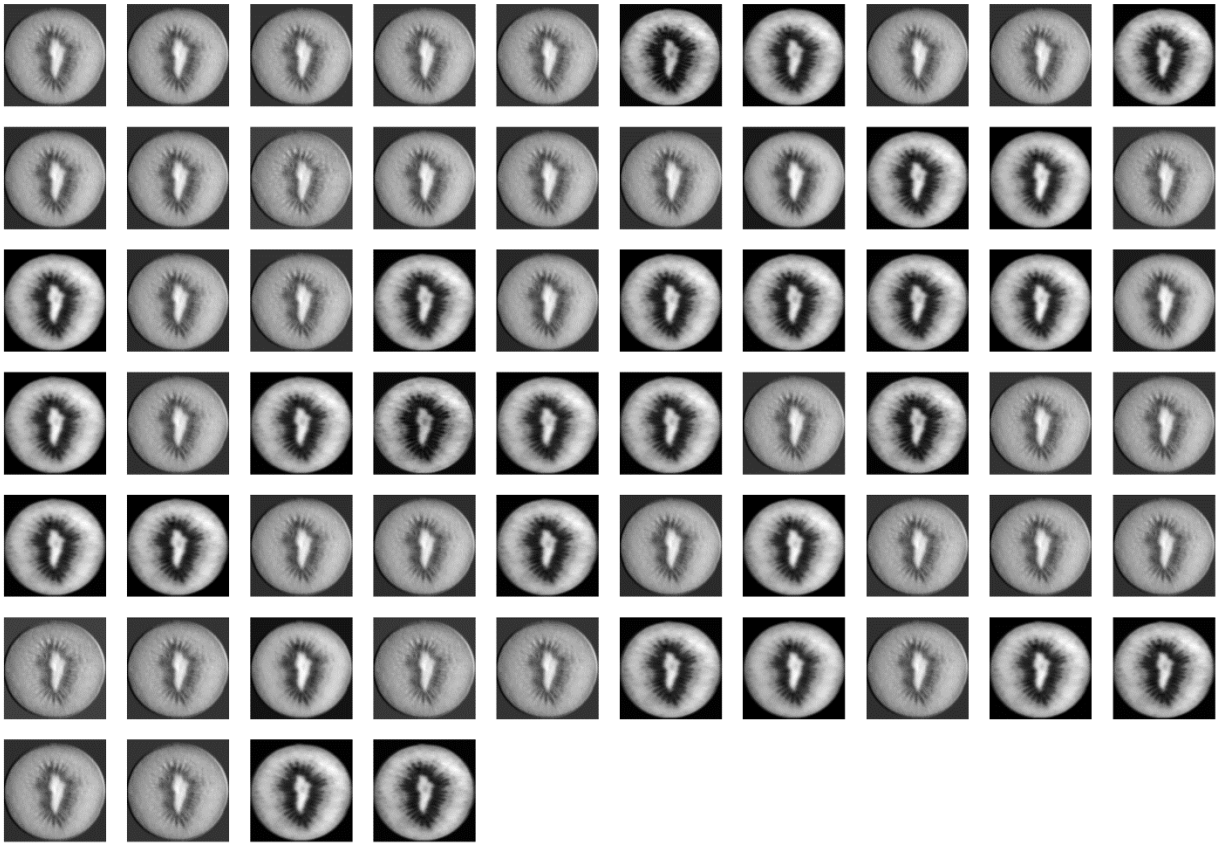


Figure 4.48: Visualisation of single kiwifruit after fifth convolution from 64 filters.

Kiwifruit at 583nm (Figure 4.51) looks similar to half of these convolved images (Figure 4.50). The other half appear to show some sort of texture feature which does not appear to obviously align to another wavelength.

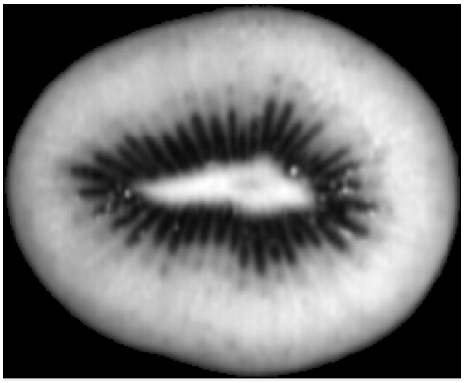


Figure 4.49: Image of a single kiwifruit at 583nm.

4.3.7 Comparison of Alternative Techniques

The above sections have explored the development of KiwiNet as a means of consistently producing a measure of Kiwifruit redness when using different cameras and under different lighting conditions. A final comparison is undertaken here to answer the question, how does KiwiNet's capability compare to simply taking an image on a typical smartphone?

When comparing 16-bit images in AdobeRGB from a Samsung and iPhone device to the redness scores produced by KiwiNet, the images produced a MAE of 7519.76 with a standard deviation of 671.72 over a random sample of 174 image pairs compared to the ground truth redness score. The ground truth being the result of subtracting the average green from the average red channel of 16bit images in AdobeRGB that were created from HS images of each kiwifruit and clipping their values between 0 and 10,000. This confirms, explicitly, that (1), as expected, KiwiNet significantly outperforms a typical consumer smartphone when measuring Kiwifruit redness, and (2), reaffirms that simply imaging Kiwifruit with a typical smartphone camera is not sufficient to measure Kiwifruit redness. Whilst the smartphones produced large errors compared to the true redness score, when comparing the iPhone and Samsung to each other, the MAE is 2556.05 with a standard deviation of 450.155. That is to say, that whilst the two devices were far from the true redness score, they were much closer to each other's redness score.

Another test was completed using the xyz2rgb colour transformation matrices in the RAW image's metadata. Over a random sample of 170 images, the iPhone produced a mean distance error of 10,035.07 from the true redness score with a standard deviation of 1349.70. Similarly, the Samsung produced a mean distance error of 11,840.35 with a standard deviation of 1516.31 from the true redness score. Comparatively, the iPhone and Samsung were on average 1805.28 from each other with a standard deviation of 215.08. There errors are consistent in direction meaning an offset may be applied. However, after a mean offset correction was applied the phones still produced MAEs as high as 6567 from the true redness value. Again, this confirms that a typical smartphone alone is not sufficient to estimate Kiwifruit Redness.

Finally, a comparison was done using an SVM rather than a NN to predict redness score from the average spectra of a kiwifruit using MATLABS *fitrsvm*. The *fitrsvm* creates a regression SVM and provides an automatic optimisation routine (The MathWorks Inc, fitrsvm, 2022). This SVM produced a mean error of 338.95 with a standard deviation of 314.39 when trained on the ground truth reflectance values. A more complex SVM could likely produce similar results to the GRNN.

5.0 Discussion

The project has proved fruitful. A system has been developed that can predict a kiwifruit's average reflectance from RAW images. This reflectance data can then be used to score red kiwifruit. This system is reasonably invariant to both differences in spectral sensitivity (different cameras) and illumination and has been demonstrated to be deployable on a typical consumer smartphone.

Similarly, a new ad hoc measure of Red19 redness has been proposed which ties human/consumer preferences to Red19's colour appearance.

5.1 Sensory Trial

Overall, the sensory study provided some insight into the threshold where the average person considers one kiwifruit a significantly different colour than another.

The study provided a total of 9630 comparisons. The results show that when two fruit are less than 1500 redness score apart participants' preference for selecting one fruit from another is noisy. That is, there does not appear to be a clear trend for fruit of score 1000 or 500 apart. This result offers some insight into what consumers may consider to be a substantial enough change in colour to warrant a change in kiwifruit grade. However, given the scale was developed in an ad hoc way and used a small sample size, a larger study is needed to validate this.

Regarding the 1000 threshold data, there is one set of data results (a set being 12 fruit compared by five different participants) where participants agree almost completely (around 98%) and this seems to fall significantly outside of the dataset's mean. However, visual inspection of this set of fruit does not present any unusual findings. Likewise, given that each set of fruit is ranked by five different participants, it is difficult to consider it an outlier. Similarly, it does not appear that one set of fruit is larger, or of differing contrast to the other. Comparatively, one set of fruit from block 7, which spans roughly the same range of redness scores, produced an agreement of only ~76%. These conflicting agreements may suggest that the metric used to create the redness score (average red minus the average green channel) may not fully explain redness as is perceived by consumers and further study is needed. This should be further explored in a larger sensory study is needed to determine what parts of red kiwifruit contribute most to perceived redness.

During the study it was decided not to resize all the kiwifruit to the same image dimensions i.e., 200 x 200px, as this would compromise the relative colour of one fruit compared to another by stretching and/or

compressing some regions of the fruit. However, in doing so, both the size and shape of each kiwifruit has likely produced noise in the results. Some participants stated that when they were unsure of which fruit to select during the experiment, it was often the size or shape of the fruit that aided their selection. Similarly, whilst the trials were prefaced with a question *to select the fruit which appears most red* to each participant, it is unclear whether this alone is sufficient to control for people's tendency to select the fruit they most want to consume or appeals to them – which may differ from which is most red.

Considering the mean and median agreement between participants, the results show a general and expected trend that as the difference in redness score decrease between kiwifruit so does agreement for which fruit is redder than another. The results suggest that fruit which are above 1500 score apart can be consistently predicted for which fruit appears most red to an average consumer.

During the process of generating AdobeRGB images, a contrast adjustment was carried out by converting each image into the CIE LAB colour space and adjusting the lightness channel. As a result of this process, some images appear to be more or less contrasted than others. There is no perfect solution to this problem. If a maximum and minimum value are used across the whole dataset, the most extreme high and low values would have to be used across every image which leads to under contrasted images for a large majority of the dataset. Given that contrast is important in perceiving the relative brightness of colours, under-contrasted images are not ideal for colour comparisons. Similarly, using the mean high and low lightness values to contrast adjust images leads to some images being over-contrasted and having saturated values and as a result, colourimetric information is lost.

During the creation of datasets for the sensory study, if a highly contrasted image was selected for incorporation into the dataset, it was manually swapped for the closest scoring neighbour of more typical contrast. There is little doubt that this contrast adjustment added some noise to the sensory trial.

It is also important to consider that the images used in this experiment were created using the reflectance data captured from the HSC. Section 3.4.3 delves extensively into the quality of this data and found that over the wavelengths 400nm-450nm, there was a high degree of instrument noise. This means that when converting from reflectance into AdobeRGB, there is some error generated by using this reflectance data particularly in the blue channel. All the images are beholden to this error and means that there is likely an offset between the absolute AdobeRGB values and the estimated ones. This should not have negatively impacted the sensory trial as the relative colours of each fruit compared to each other are correct.

Whilst this project used highly varied images of kiwifruit to assess redness, a simpler approach would be to control the size and shape of kiwifruit using a standard kiwifruit image template and varying only the RGB values of each segment of kiwifruit. This would allow the relative contribution of the inner, outer, and core segments of the kiwifruit to be more precisely determined. This may also be extended to determine a just noticeable difference or ΔE for red kiwifruit. Such an experiment would provide a more definitive and uni-dimensional approach to measuring perceivable colour difference although what is defined as perceivable colour difference has been already been found to range from 2.3 (Cugmas & Struc, 2020) – 6 (Cugmas, Pernus, & Likar, 2017).

It is important to remember that it is not the intention, nor within the scope, of this project to extensively validate a new measure of Red19 kiwifruit redness using consumer preference data. Rather, this small sensory trial allowed the project to use a basic redness scale for training purposes that shows promise in relating to actual consumer preferences. The validation of such a scale is another project in and of itself.

One shortcoming of the sensory trial was that the opportunity was not taken to also collect consumer data using Zespri's original redness scale as well as the new scale. This would have enabled a comparison between the ad hoc scale with the current redness scale where corelations could be explored. This would be wise in future research. Similarly, the ISO standard for such sensory trials recommend that 100 participants are needed to successfully validate the trial although (Kemp, Hollowood, & Hort, 2009) suggest a sample of 30 is sufficient. This study had 34 participants.

Another approach that may have proven useful in establishing a scale of redness is that of Wang & Zhang (2013) who defined the colour gamut of human tongues before creating a custom CCT to include in image comparisons. Mapping the complete gamut of Red19 kiwifruit may greatly assist in developing a scale to span its range.

5.2 CNN

The initial results show that calculating network loss using RMSE appears to produce the lowest errors compared to L1 and MSE loss. This is in agreement with many other studies (Zhao, et al., 2020; Racek & Baláž, 2010). When adding a weighting to the loss function based on the variance explained by each principal component, there was no improvement in network error. However, after reconstructing the full spectra and passing it to the loss function there was significant improvement in network performance. Calculating loss using the reconstructed spectra also serves as a way of weighting each principal component although this clearly worked better than using the variance they each explain.

The comparison of the visible and NIR spectral ranges do not appear to produce significantly different results, and this trend appears to hold throughout the rest of the analysis. This would imply that the NIR spectral range does not hold useful information that relates to the colour quality of Red19 Kiwifruit. It should be acknowledged that the redness scale was created in AdobeRGB colour space which naturally does not contain any information in the NIR spectral range. Moreover, the average reflectance of Red19 kiwifruit appear to be similar in the NIR range across all the classes of kiwifruit and this would suggest that NIR does not contain significant information pertaining to fruit colour.

In regard to the estimation of reflectance, the CNN produced an RMSE of 0.0131 and 0.0113 over the visible and NIR spectral ranges. It is difficult to compare these results with current state-of-the-art methods as most SSR networks are trained to resolve entire images and use deep network architectures. There is little doubt that implementing a network as deep as the Marianas trench will produce superior results, although it is unlikely network inferences could be made on regular smartphones given the memory required. During training, the CNN required approximately 870 Mb of GPU memory and for a single inference the network uses approximately 0.46 Mb of memory. State-of-the-art models like that of Shi et al., (2018) use between 12GB and 32GB of memory for a single inference depending on the network. Given the lightness of the KiwiNet model, it is suitable for simple and quick use on a typical smartphone. Constraints of local memory may be circumvented by implementing network inferences on third party hardware using cloud services and such systems are not uncommon.

During the training of the CNN when predicting redness score and kiwifruit class, grower 7174A showed a significantly higher failure rate than other growers. Even after augmenting the data and ensuring the network was invariant to augmented data, this grower still had a high error rate. Grower 7174a's data was collected over three separate days. It does not appear visually different from other fruit and its scores do not appear to fall overly close to classification boundaries.

In comparison, when predicting redness score, grower 7174a's error is only slightly higher than the network average and appears to fall within the expected range. Similarly, the reflectance error of grower 7174a is again, only slightly higher than the network average. This may suggest that the error only appears when training the network to estimate class although it is unclear exactly what causes this large error. Future research should take care when estimating kiwifruit class directly as the cause of this error is unknown. Estimating redness score provided much more information regarding where in a kiwifruit class a single fruit sits and it is recommended that this approach be maintained in future work.

One approach to exploring this error is class activation mapping (CAM). CAM provides a heatmap showing which spatial regions of the input image the CNN used to identify its class (Zhou et al., 2015). CAM may offer a way to understand which regions of the kiwifruit are most relevant to predicting its class and may offer insight into why grower 7174a shows higher than average errors.

Finally, whilst the kiwifruit images were rescaled before being fed into the network, each fruit had a black background. Given that fruit varied in size, the outer perimeter of the fruit would have black pixels for some images and actual fruit pixels in others. Adding a pre-processing step to address this such as unravelling is recommended for future models.

5.3 GRNN

When exploring the average spectra of different classes of kiwifruit there appears to be a trend where the higher classes (redder fruit) have a lower reflectance through the spectral range of 400nm – 650nm (figure 4.12) compared to the lower classes (greener fruit) and that these classes appear to cross over around 650nm. Considering figure 4.13, it appears that the outer pericarp is most responsible for a dip in reflectance spectra around 680nm. This dip also coincides with the absorption wavelength of chlorophyll (Guidi, Tattini, & Landi, 2017), where greener fruit appear to have a much larger dip in this region. It is expected that green fruit naturally contain more chlorophyll which absorbs red wavelengths of light and reflects green wavelengths of light.

This region between 500nm to 700nm appears to show the most sensitivity to changes in kiwifruit colour and these wavelengths are primarily the red and green spectra of light. Blue light appears to show a small interaction between average reflectance and redness score (Figure 3.15) and this would suggest that kiwifruit redness is not strongly correlated to blue wavelengths of light.

Considering the architecture of the GRNN, the size and shape of the pattern recognition and summation layers performed best with all the layers being of equal size and when the pattern recognition layer was double the input layer. The network with a smaller summation layer performed slightly worse than the others, although given the standard deviations of the errors are around 150 on average, it cannot be claimed that the traditional configuration performs significantly worse.

When comparing the predictive power of different segments of kiwifruit, it appears that both the average of the whole fruit and the outer pericarp region have the same predictive power regarding the redness score of a kiwifruit. It would be expected that the network with the most amount of information in its input layer would perform the best even if the amount of additional colour information possessed by the additional inputs is only a small amount. However, when the network has all four segments, it does not perform significantly better than when trained on the whole or outer segments alone. Considering figure 4.15, the RMSE values appears to reach a minimum around the 50th epoch and then vary by up to ~400 as training continues. This may suggest that the network's learning rate is too high, although the results from the hyperparameter tuning does not support this, or that the average reflectance does not contain *all* of the necessary data to predict redness score and thus produces a noisy error. This error may be alleviated by

increasing the spatial resolution of this data. There is always a natural trade-off between network size and predictive power. Future research exploring the optimal balance between the prediction of redness score vs memory usage would be beneficial.

The core segment of kiwifruit showed a failure rate of ~60%. It is expected that the core of each fruit, which is essentially a completely white image, would be misclassified around 80% of the time given an equal guess between 5 classes. Whilst still poor, this suggests that there is some information within the core segment that explains redness score.

Likewise, the inner segment appears to misclassify fruit around ~25% of the time which suggests that it does contain some amount of useful information regarding redness score. However, given that when the outer and inner segments were combined, it produced almost identical errors it would suggest that the colour information in the inner segment is also explained in the outer segment, making the inner segment redundant regarding the information it provides.

Considering that the images were given a redness scored based on the difference of red and green colour channels in the outer pericarp region, it is not a particularly startling finding to see the inner pericarp does not contain additional information from the outer pericarp. However, given that the redness score was created as to agree, albeit somewhat approximately, with consumer preferences of kiwifruit redness, this may suggest that redness of kiwifruit is derived primarily from the outer pericarp region and less emphasis can be given to the inner pericarp region when grading Red19 kiwifruit. Likewise, it may suggest that the current metric of 'redness spread' may also not be equally important compared to the redness of the outer pericarp.

In regard to the optimisation of the GRNN's hyperparameters, the sweeps show that the visible spectral range, learning rate, and dropout percentage are positively correlated with network error. Given that much of the findings in section 4.3.2 do not seem to show a clear preference for which spectral range explains redness better than the other and the corelation between spectral range and network error is weak, this is likely the result of noisy data.

With regard to layer size, initial results show that a summation layer first, then a pattern recognition layer produced lower network errors than a more traditional GRNN architecture although after a larger set of sweeps was completed, the results show a larger pattern recognition and larger summation layer produced a lower network error. Considering that a summation layer preceding a pattern recognition layer is akin to reading half a book, throwing away the other half and then guessing the ending, it is not likely that these results are conclusive and given that the number of sweeps was only 300, further testing is required. A more expected result appears in the final results after 700 sweeps were completed where both layers where negatively correlated with network error. This suggests, as expected, that as the size of the pattern recognition and summation layers increase, the network error decreases.

5.4 Synthetic Dataset Generation

Comparing synthetic and real RAW images, when the network is trained and tested on synthetic data, it appears to produce significantly lower network errors than when trained and tested on real data. It is expected that data that has been procedurally generated would contain patterns that the network may be able to exploit compared to data captured in the real world. This may explain why the network performs better when trained and validated on synthetic data.

Visually, the real and synthetic RAW data from the iPhone 12 Pro and Pro Max synthetic data appear similar to the eye (Figure 4.41).

When combining the real and synthetic images, the resulting RMSE falls between the values found when using only the real or only the synthetic data for training as expected. Comparing the error between real and synthetic images, when training the network on the combined dataset, the network produces roughly twice the average errors on the real data compared to the synthetic data. This would suggest that there is some colour information in the real data that is not explained in the synthetic data. However, when training the network on synthetic data then and validating it on real data and vice versa, both instances produce very similar errors.

Given that when this experiment was undertaken care was given to the rotation and lateral inversion of the images, it may suggest that the KiwiNet network is not robust to rotation and lateral inversion.

Due to the methods used, the smartphone images and the reflectance data naturally fall at right angles to each other and when the network is trained using the combined datasets, the real and synthetic images could be explained by the polarity of the kiwifruit. That is, real vs synthetic data can be identified by the axes that the major axis of the kiwifruit cross sections align to.

During the augmentation of data (section 4.3.1) the network showed it was invariant a 180° rotation and lateral inversion of the images. However, a 90° rotation was not explored, and this would explain the higher errors on the real images produced by the combined dataset. It is also worth noting that the real and synthetic data appear in a ratio of 6927:5000 thus, the network has more real than synthetic data to learn from. Given the similarity in error when rotation and lateral inversion are controlled, this is still likely the cause of this error. Exploring further augmentation operations as well as exactly how much real data is needed to sufficiently train a model may prove useful in future work.

Training invariance to rotation is a reasonably trivial task and can be achieved by randomly propagating images of rotated kiwifruit to feed into the network. Thereafter, it is simply a matter as to how much data is needed before the network learns invariance to rotation. Training the model for invariance to rotation should be completed before testing the mobile app in the industry.

Finally, it is worth noting that specular reflections are reasonably common in the HS data and the resultant AdobeRGB images. Specularities are a sign of sensor saturation and do not hold any meaningful

colourimetric information. Due to the perishable nature of kiwifruit and ongoing supply chain issues globally due to Covid-19, there was simply not enough time to acquire the needed light diffusing equipment to sufficiently reduce these specularities. Their effect is likely minimal and would manifest by increasing the average RGB channels due to their high values. Care should be taken to prevent specularities in future research.

5.5 KiwiNet

When combining the CNN and GRNN networks and training the network to minimise the error in redness score, the estimation of kiwifruit reflectance does not align with the ground truth reflectance data at all. However, the resulting pseudo-reflectance that is generated does present some curious patterns. Namely, when aligned with the wavelengths of light that the original reflectance values represented, the pseudo-reflectance values show that between the 400nm-500nm range there is a separation between kiwifruit classes with the higher classes having lower values in this range. Comparatively, it appears that the pseudo-reflectance is most responsive to different images between the ranges of 520nm-680nm. Between these ranges, reflectance values follow a parabolic-like path and reach a maximum value at 583nm. This may suggest that the value of this wavelength may contain the most important information pertaining to the estimation of redness score.

However, considering that the pseudo-reflectance is the result of a back propagation over three dense layers and a single convolutional layer, it cannot be assumed that the pseudo-reflectance is still represented in the wavelength space. There does appear to be a relationship (Figure 4.44) but more research is needed to confirm these findings. Further research exploring the maximum point (583nm) of this pseudo-reflectance may offer a way to reduce redness score to a single point value which has significant benefits in terms of simplicity and interpretability. If this wavelength does contain the best representation of redness score, a larger, full-image network that predicts the entire kiwifruit at this wavelength may offer a fruitful balance between spectral and spatial information.

When training the CNN and GRNN parts of KiwiNet separately and then measuring the resultant errors, the network produced lower error when each sub-network was train separately compared to when they are trained together. Separate training resulted in a decrease in absolute error by approximately 100 for the mean, median, and standard deviation of the redness score errors. These findings may also suggest that the real kiwifruit reflectance provides more useful information to the GRNN than the pseudo-reflectance as the network errors are lower when estimating score from true reflectance spectra.

Finally, a proof-of-concept mobile application has been created that proves that both the CNN and GRNN models can be deployed on a typical smartphone. Because Red19 kiwifruit is not currently in season at the time of this writing, validation tests of this application could not be completed. The expected accuracy of the application's reflectance estimation is 0.0109 and 0.0096 RMSE over the visible and visible NIR spectral ranges. Similarly, the prediction of kiwifruit redness score has a mean absolute error of 220.33 with a standard deviation of 213.82 which is equivalent to an average error of 2.2% and standard deviation of 2.13% when considering the full range of scores on the redness scale. The combined result of these accuracies produced a correct kiwifruit classification rate of 91.06%.

6.0 Conclusions & Recommendations

In conclusion, this project set out to explore and assess how to objectify the assessment of Red19 kiwifruit colour quality. This project has shown that by using a single convolutional layer and three dense layers, the average reflectance of Red19 kiwifruit can be estimated reliably from RAW images across a range of smartphones and illuminants.

Similarly, this project has proposed and undertaken a small validation study of a new scale to measure the redness of Red19. This study showed that when two kiwifruit are compared which score at least 1500 apart on this new redness scale, people will almost always agree which fruit is redder than the other. This finding may offer a way of classifying Red19 kiwifruit in relation to consumer preferences although a larger study is required to better validate this scale.

This project has also shown that a GRNN can be applied to solve the issue of regressing from the average reflectance of a kiwifruit to its redness score and that this method performs better than simple comparisons of RAW/sRGB data and outperforms a basic SVM. A new NN has been created, named KiwiNet, that effectively combines the CNN and GRNN models proposed in this project. KiwiNet has shown a successful classification rate of between 91.05% across ~11,000 RAW images taken across 12 smartphones and under 5 different lighting conditions. KiwiNet needs as little as 0.87 MB of memory for training and only ~0.5Mb for inference making it suitable for local usage on modern day smartphones.

This study can conclude that there does not appear to be a significant difference between using visible spectra or visible NIR spectra when estimating the redness score of kiwifruit. Likewise, using the average spectra of the whole kiwifruit or outer pericarp region provides the best estimates of redness score. The inner and core regions of kiwifruit do not appear to provide additional information related to a kiwifruit's redness score. These results suggests that consumers do not equally weight the contribution of the redness of the inner and outer pericarp regions in relation to overall kiwifruit redness. Considering the addition of a weighting to scores based on kiwifruit region may assist in improving Zespri's current scoring system.

KiwiNet showed that the 583nm wavelength appeared to provide a single point measure of kiwifruit redness. 583nm is a yellow signal of light in the visible range and it is likely that this wavelength represents the ad hoc redness scale that was created by subtracting the average red and green channel of the outer pericarp region of kiwifruit as the mixing of green and red light produce yellow. Hence, the intensity of this yellow may offer a consistent and reliable way to assess Red19 redness. The 583nm wavelength of light should be further investigated in relation to the estimation of redness score and future research estimating a single image at this wavelength is recommended.

Likewise, a larger sensory trial should be carried out that controls for different colours in different regions of kiwifruit. This study should also have participants rate kiwifruit on a/the new redness scale *as well as* Zespri's existing scale so that corelations can be explored. Mapping the colour gamut of Red19 Kiwifruit is also advisable as it allows for the full range of kiwifruit colour to be known and a scale created to span this range.

Finally, convolutional activation mapping is recommended to further understand which regions of the kiwifruit are most important to the model when reaching its decisions and this information will prove useful should a similar system be used for grading fruit in industry. This would also offer kiwifruit growers and breeders a means to better understand which areas of the kiwifruit are most desirable to consumers but only if the redness scores are determined using a large and more extensive sensory trial.

7.0 References

Adobe. (2021). Adobe Lightroom 5.1. *https://www.adobe.com/nz/products/photoshop-lightroom.html*.

Adobe Systems Incorporated. (2005, May). Retrieved from Adobe.org: https://www.adobe.com/digitalimag/pdfs/AdobeRGB1998.pdf

Aeschbacher, J., Wu, J., & Timofte, R. (2017). In defense of shallow learned spectral reconstruction from RGB images. *2017 IEEE International Conference on Computer Vision Workshops* (pp. 471-479). IEEE.

Afifi, M. (2018). Semantic white balance: semantic color constancy using convolutional neural network. *arXiv*, 1802.00153.

Afifi, M., Baron, J. T., LeGendre, C., Tsai, Y.-T., & Bleibel, F. (2021). Cross-Camera Convolutional Color Constancy. *The IEEE International Conference on Computer Vision (ICCV)*.

Akhtar, N., & Mian, A. (2018). Hyperspectral recovery from RGB images using gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (pp. 42(1), 100-113). IEEE.

Akkaynak, D., Treibitz, T., Xiao, B., Gurkan, U. A., Allen, J. J., Demirci, U., & Hanlon, R. T. (2014). Use of commercial off-the-shelf digital cameras for scientificdata acquisition and scene-specific color calibration. *Optical Society of America*, 31(2)" 312-321.

Albawi, S., Mohammed, T. A., & Al-Zaw, S. (2017). Understanding of a convolutional neural network. *2017 International Conference on Engineering and Technology (ICET)* (pp. 1-6). IEEE.

Alvarez-Gila, A., Weijer, J. V., & Garrote, E. (2017). Adversarial Networks for Spatial Context-Aware Spectral Image Reconstruction from RGB. *IEEE International Conference on Computer Vision Workshops (ICCVW)* (pp. 480-490). IEEE.

Amani, M., Falk, H., Jensen, O. D., Vartdal, G., Aune, A., & Lindseth, F. (2019). Color calibration of human skin images in proceedings of the computer vision systems. *International Conference on Computer Vision Systems* (pp. 211-223). Cham, Switzerland: Springer International Publishing.

Apple Inc. (2020). iPhone 12 Pro. *https://www.apple.com/nz/iphone-12/key-features/*.

Arad et al. (2020). NTIRE 2020 Challenge on Spectral Reconstruction from an RGB Image. *CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (pp. 1806-1822). IEEE.

Arad, B., & Ben-Shahar, O. (2016). Sparse recovery of hyperspectrak signal from natural RGB images. *ECCV*.

Avilaa, F., Moraa, M., Oyarcea, M., Zuñigaa, A., & Fredes, C. (2015). A method to construct fruit maturity color scales based on supportmachines for regression: Application to olives and grape seeds. *Food Engineering*, 162: 9-17.

Aznan, A., Viejo, C. G., Pang, A., & Fuentes, S. (2021). Computer Vision and Machine Learning Analysis ofCommercial Rice Grains: A Potential Digital Approach forConsumer Perception Studies. *Sensors*, 21: 6354.

Baherinia, H., & Manduchi, R. (2011). Robust real-time detection of multi-color markers on a cell phone. *Real-Time Image Processing*, 8:207-223.

Balasubramanian, R. (2003). Device charactierization. In G. Sharma, *Digital color imaging handbook* (pp. 271 - 383). Boca Raton, Florida, USA: CRC Press LLC.

Barron, J. T. (2015). Convolutional Color Constancy. *arXiv: 1507.0.00410*.

Bianco, S., Cusano, C., & Schettini, R. (2015). Single and Multiple Illuminant Estimation Using Convolutional Neural Networks. *2015 Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (pp. 81-89). IEEE.

Biewald, L. (2022, Jan 20). *Experiment Tracking with Weights and Biases*. Retrieved from Weights & Biases: https://www.wandb.com/

Can, Y. B., & Timofte, R. (2018). An efficient CNN for spectral reconstruction from RGB images. *arXiv*, 1804.04647.

Carrio, A., Sampedro, C., Sanchez-Lopez, J. L., Pimienta, M., & Campoy, P. (2015). Automated low-cost smartphone-based lateral flow saliva test reader for drugs-of-abuse detection. *Sensors*, 15: 29569-29593.

Carvalho, D. V., Pereira, E. M., & Cardoso, J. S. (2019). Machine learning interpretability: a survey on methods and metrics. *Electronics*, 8(8), 832.

Chakrabarti, A., & Zickler, T. (2011). Statistics or real-world hyperspectral images. *Proceedings / CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 193-200). IEEE Computer Society Conference on Computer Vision and Pattern Recognition: IEEE .

Charriere, R., Hebert, M., Tremeau, A., & Destouches, N. (2013). Color calibration of an RGB camera mounted in front of a microscope with strong color distortion. *Applied Optics*, 52, 5262-5271.

Chen, X., Drew, M. S., Z.-N. L., & Fraser, S. (2017). Illumination and reflectance spectra seperation of hyperspectral image data under multiple illumination conditions. *Electronic Imaging, Color Imaging XXII: Displaying, Processing, Hardcopy, and Applications, pp. 194-199(6)* (pp. 6,194-199). Society for imaging science and technology.

Chen, Y., Fu, G., Zilberman, Y., Ruan, W., Ameri, S. K., Zhang, Y. S., . . . Sonkusale, S. R. (2018). Low cost smart phone diagnostics for food using paper-based colorimetric sensor arrays. *Food Control*, 27: 227-232.

Chen, Z., & Wang, K. (2018). GROF: indoor localiztion using multiple-bandwidth general regression neural networks and outlier filter. *Sensors*, 18: 3723.

Chen, Z., Huang, H., & Liao, N. (2018). The color metamerism evaluation of paint based on ocean spectrum. *IOP Conference Series: Materials Science and Engineering.* (p. 322). IOP Publishing.

Choi, K., Lee, J., & Suk, H.-J. (2016). Context-based presets for lighting setup in residential space. *Applied Ergonomics*, 52, 222-231.

CIE. (2018). *CIE 1931 colour-matching functions , 2 degree observer (data table),*. Vienna, Austria, DOI:10.25039/CIE.DS.xvudnb9b: International Commission on Illumination (CIE),.

Clark, E. (2021, June 2). *Cuttings of prized SunGold kiwifruits were smuggled to China and NZ growers are divided over what to do about it*. Retrieved from ABC news: https://www.abc.net.au/news/2021-06-02/china-new-zealand-kiwifruit-trade-impacts-/100129232

Clark, J. H. (1924). The Ishihara Test for Color Blindness. *American Journal of Physiological Optics*, 5, 269-276.

Cugmas, B., & Struc, E. (2020). Accuracy of an affordable smartphone-based teledermoscopy system for color measurements in canine skin. *Sensors*, 20:6234.

Cugmas, B., Pernus, F., & Likar, B. (2017). Color constancy in dermatoscopy with smartphone. *Proc. SPIE 10592, Biophotonics—Riga 2017.*

Deng, J., Dong, W., Socher, R., Li, L.-J., li, K., & Fei, L.-F. (2009). ImageNet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 248-255). IEEE.

Dogan, V., Yuzer, E., Kilic, V., & Sen, M. (2021). Non-enzymatic colorimetric detection ofhydrogen peroxide using aμPAD coupled with amachine learning-based smartphone app. *Analyst*, 146, 7336-7344.

Dong, C., Loy, C. C., He, K., & Tang, X. (2016). Image super-resolution using deep convolutional networks. *IEEE Transactions on pattern analysis and machine intelligence* (pp. 38(2):295-307). IEEE .

Dorrepaal, R., Malegori, C., & Gowen, A. (2016). Tutorial: time series hyperspectral image analysis. *Near Infrared Spectroscopy*, 24: 89-107.

EIZO Corportation. (2016). ColorEdge CG2420. *https://www.eizoglobal.com/products/coloredge/cg2420/*.

ElMasry, G., & Sun, D.-W. (2010). Principles of hyperspectral imaging technology. In D.-W. Sun, *Hyperspectral Imaging for Food Quality Analysis and Control* (pp. 3-43). Academic Press.

Elvidge, C. D., Keith, D. M., Tuttle, B. T., & Baugh, K. E. (2010). Spectral Identification of Lighting Type and Character . *Sensors*, 10: 3961-3988.

Fairchild, M. D. (2014). Seeing, adapting to, and reproducing the apperanace of nature. *Applied Optics*, 54(4), 107-116.

Fankhauser, N., Kalberer, N., Muller, F., Leles, C. R., Schimmel, M., & Srinivasan, M. (2020). Comparison of smartphone-camera and convential flatbed scanner images for analytical evaluation of chewing function. *Oral Rehabilitation*, 47: 1496-1502.

Fernández-Delgado, M., Sirsat, M., Cernadas, E., Alawadi, S., Barro, S., & Febrero-Bande, M. (2019). An extensive experimental survey of regression methods. *Neural Networks*, 111: 11-34.

Flachot, A., Akbarinia, A., Schutt, H. H., Fleming, R. W., Wichmann, F. A., & Gegenfurtner, K. R. (2020). Deep neual models for color discrimination and color constancy. *Journal of Vision*.

Foster, D. H., & Amano, K. (2019). Hyperspectral imaging in color vision research: tutorial. *Optical Society of America*, 36(4): 606-627.

Fourure, D., Emonet, R., Fromont, E., Muselet, D., & Tremeau, A. (2016). Mixed pooling neural networks for color constancy. *2016 IEEE International Conference on Image Processing* (pp. 3997-4001). Phoneix, Arizona, USA: IEEE.

Fu, Q., Zhang, C., Xie, J., Li, Z., Qu, L., Cai, X., . . . Tang, Y. (2019). Ambient light sensor based colorimetric dipstick reader for rapid monitoring organophosphate pesticides on a smart phone. *Analytical Chimica Acta*, 1092: 126-131.

Funt, B., Cardei, V., & Barnard, K. (1998). Learning color constancy. *The Fourth Color Imaging Conference: Color Science, Systems and Applications* (pp. 58-60). Scottsdale, Arizona, USA: Society for Imaging Science & Technology .

Galliani, S., Charis Lanaras, Marmanis, D., Baltsavias, E., & Schindler, K. (2017). Learned spectral super-resolution. *arXiv*, 1703.09470.

Garcia, A., Erenas, M. M., Marinetto, E. D., Abad, C. A., Orbe-Paya, I. d., Palma, A. J., & Capitan-Vallvey, L. F. (2011). Mobile phone platform as portable chemical analyzer. *Sensors and Actuators B: Chemical*, 156: 350-359.

GlobeNewswire. (2020, Feburary 05). *Kiwi Fruits Market Report 2020: Industry Value & Volume, Trends, Production & Consumption, Imports & Exports, Prices, and Key Market Players Through 2007-2025*. Retrieved from GlobeNewswire: https://www.globenewswire.com/news-release/2020/02/05/1980342/0/en/Kiwi-Fruits-Market-Report-2020-Industry-Value-Volume-Trends-Production-Consumption-Imports-Exports-Prices-and-Key-Market-Players-Through-2007-2025.html

Goni, S. M., & Salvadori, V. O. (2017). Color measurement: comparison of colorimeter vs. computer vision system. *Food Measurement*, 11: 538-547.

Goni, S. M., & Salvadori, V. O. (2017). Color measurement: comparison of colorimeters vs computer vision system. *Food Measurement and Characterization*, 11, 538-547.

Griciuc, A. (2010). Molecular and functional analysis of the ERAD effector VCP in cellular and Drosophila models for retinitis pigmentosa.

Guidi, L., Tattini, M., & Landi, M. (2017). How Does Chloroplast Protect Chlorophyll Against Excessive Light? In InTech, *Chlorophyll* (pp. 21-36). InTech.

Hardeberg, J. Y. (2002). On the spectral dimensionality of object colours. *Conference on colour in graphics, imaging, and vision vol. 2002* (pp. 480-485). Society for imaging science and technology.

Hasan, R., & Monir, S. M. (2017). Fruit maturity estimation based on Fuzzy Classification. *IEEE International Conference on Signal and Image Processing Applications (IEEE ICSIPA 2017)* (pp. 27-32). Malaysia: IEEE.

Hoffmann, G. (2021, Dec 24). *CIE color space.* Retrieved from Gernot Hoffmann in Emden / Germany: http://docs-hoffmann.de/howww41a.html

Hordley, S. D. (2006). Scene Illuminant Estimation: Past,Present, and Future. *Color Res. Appl.*, 31: 303-314.

Hu, Y., Wang, B., & Lin, S. (2017). FC^4: Fully Convolutional Color Constancy with Confidence-Weighted Pooling. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 330-339). IEEE.

Huang, G., Liu, Z., & Maaten, L. v. (2018). Densely Connected Convolutional Networks. *arXiv*, 1608.06993.

Hunt, D. M., & Carvalho, L. S. (2016). The genetics of color vision and congenital color deficiencies. In J. Kremers, R. C. Baraas, & N. J. Marshall, *Human color vision* (pp. 1-32). Cham, Switzerland: Springer International Publishing.

Hunt, R., & Pointer, M. (2011). *Measuring Color.* West Sussex, England: John Wiley & Sons Ltd.

Image Analyst. (2022, Feb 20). *MATLAB Central File Exchange*. Retrieved from Mathworks.com: https://www.mathworks.com/matlabcentral/fileexchange/38236-color-checker-chart

Inoue, K., Hara, K., & Urahama, K. (2017). Spectral reflectance estimation and color reproduction based on sparse neugbauer model. *Advances in Science, Technology and Engineering Systems*, 2(3)" 658-966.

International Commission on Illumination. (2019-06). Colorimetry - part 1: CIE standard colormetric observers. In I. O. Standardization. Geneva, Switzerland: International Organization for Standardization.

Ishihara, S. (1972). *Tests for colour-blindness.* Tokyo, Japan.

Ives, F. E. (1990). The optics of trichromatic photography. *Photographic Journal*, 40, 99-121.

Jegou, S., Drozdzal, M., Vazquez, D., Romero, A., & Bengio, Y. (2017). The one hundred layers tiramisu: fully convolutional densenets for semantic segmentation. *arXiv*, 1611.09326.

Jiang, J., Liu, D., Gu, J., & Susstrunk, S. (2013). What is the space of spectral sensitivity functions for digital color cameras. *2013 IEEE Workshop on Applications of Computer Vision (WACV)* (pp. 168-179, doi: 10.1109/WACV.2013.6475015.). IEEE.

Johnson, G. M., & Fairchild, M. D. (2003). Visual psychophysics and color apperance. In G. Sharma, *Digital color imagine handbook* (pp. 115-172). Boca Raton, Florida, USA: CRC Press LLC.

Kamper, H. (2017). Yet another introduction to backpropagation.

Kap, O., Kilic, V., Hardy, J. G., & Horzum, N. (2021). Smartphone-based colorimetric detection systems for glucose monitoring in the diagnosis and management of diabetes. *Analyst*, 146, 2784-2806.

Karlsen, H., & Dong, T. (2015). A smart-phone based robust correction algoithm for the colorimetric detection of urinary tract infection. *Annu Int Conf IEEE Eng Med Biol Soc* (pp. 1251-1254). IEEE.

Kaur, H., Sawhney, B. K., & Jawandha, S. K. (2018). Evaluation of plum fruit maturity by image processing techniques. *Food Sci Technol*, 55(8), 3008-3015.

Kemp, S. E., Hollowood, T., & Hort, J. (2009). *Sensory evaluation a practical handbook .* West Sussex, UK: John Wiley & Sons Ltd.

Keras. (2022a, Feb 19). *Model training APIs*. Retrieved from Keras: https://keras.io/api/models/model_training_apis/#:~:text=Number%20of%20samples%20per%20batch,dataset%2C%20generators%2C%20or%20keras.

Keras. (2022b, Feb 19). *Adam*. Retrieved from Keras: https://keras.io/api/optimizers/adam/

Kılıç, V., Horzum, N., & Solmas, M. E. (2018). From sophisticated analysis to colormetric determination: smartphone spectrometers and colorimetry. In *Color Detection.* IntechOpen.

Kim, H., Awofeso, O., Choi, S., Jung, Y., & Bae, E. (2017). Colorimetric analysis of saliva-alcohol test strips by smartphone-based instruments and using mahcine-learning algorithms. *Applied Optics*, 56(1): 84-92.

Kim, M. S., Chen, Y. R., & Mehl, P. M. (2001). Hyperspectral reflectance and flurorescence imaging system for food quality and safety. *Transactions of the ASAE* (pp. 44(3): 721-729). American Society of Agricultural and Biological Engineers.

Kortum, G. (1969). *Reflectance Spectroscopy: principles, methods, applications.* New York City, USA: Springer-Verlag New York Inc.

Koščević, K., Banic, N., & Loncaric, S. (2019b). Color Beaver: Bounding Illumination Estimations for Higher Accuracy. 183-190.

Koscevic, K., Subasic, M., & Loncaric, S. (2019a). Attention-based convolutional neural network for computer vision color constancy. *11th International Symposium on Image and Signal Processing and Analysis (ISPA)* (pp. 372-377). Dubrovnik, Croatia: IEEE.

Krizhevsky, A., Sutskever, I., & Hilton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *NIPS'12: Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1* (p. 3328). Red Hook, NY, USA: Curran Associates Inc.

Kwan, C., Chou, B., & III, J. F. (2019). Comparison of deep learning and conventional demosaicing algorithms for mastcam images. *Electronics*, 8, 308.

Lazaro, A., Boada, M., Villarino, R., & Girbau, D. (2019). Color measurement and analysis of fruit with a battery-less NFC sensor. *Sensors*, 19, 1741.

Lee, B. B., & Silveira, L. C. (2016). Cone Opponency: An Efficient Way of Transmitting Chromatic Information. In J. Kremers, R. C. Baraas, & N. J. Marshall, *Human Color Vision* (pp. 105-132). Cham, Switzerland: Springer International Publishing".

Lee, B., & Silveira, L. (2016). Cone Opponency: An Efficient Way of Transmitting Chromatic Information. In J. Kremers, R. Baraas, & N. Marshall, *Human Color Vision. Springer Series in Vision Research* (pp. Vol 5, 105-132). Cham: Springer.

Leow, M. E., Ow, R. K., Lee, M. H., Huak, C. Y., & Pho, R. W. (2006). Assessment of colour differences in silicone hand and digit prostheses: perceptible and acceptable thresholds for fair and dark skin shades. *Prosthetics & Orthotics International* , 30(1):5-16.

Li, B., Wang, J., Tu, H., Yang, Z., Zhao, D., Feng, H., & Yang, J. (2020). A self-designed versatile and portable sensing device based on smart phone for colorimetric detection. *Analytical and Bioanalytical Chemistry*, 413:533-541.

Lillicrap, T. P., Santoro, A., Marris, L., Akerman, C. J., & Hinton, G. (2020). Backpropogration and the brain. *Nature*, 21: 335-346.

Lopez-Ruiz, N., Curto, V. F., Erenas, M. M., Benito-Lopez, F., Diamond, D., Palma, A. J., & Captain-Vallvey, L. F. (2014). Smartphone-based simultaneois pH and nitrite colorimetric determination for paper microfluidic devices. *Analytical Chemistry*, 86, 9954-9562.

Luther, R. T. (1927). Aus dem Gebiet der Farbreizmetrik. *Zeitschrift für technische Physik*, 8: 540-558.

Magnusson, M., Sigurdsson, J., Armansson, S. E., Ulfarsson, M. O., Deborah, H., & Sveinsson, J. R. (2020). Creating RGB images from hyperspectral images using a color matching function. *2020 IEEE International Geoscience and Remote Sensing Symposium* (pp. 2045-2048). IEEE.

Mahesh, B. (2020). Machine learning algorthims - a review. *International Journal of Science and Research (IJSR)*, 9(1): 381-386.

Manley, M. (2014). Near-infrared spectroscopy and hyperspectral imaging: non-destructive analysis of biological materials. *Chemical Society Reviews*, 43: 8200-8214.

Mariakakis, A., Banks, M. A., Phillipi, L., Yu, L., Taylor, J., & Patel, S. N. (2017). BiliScreen: Smartphone-Based Scleral Jaundice Monitoring for Liver and Pancreatic Disorders. *Proc. ACM Interactive, Mobile, Wearable Ubiquitous Technol*, 1(2), 1-26.

Mathaweesansurn, A., Maneerat, N., & Choengchan, N. (2017). A mobile phone-based analyzer for quantitative determination of urinary albumin using self-calibration approach. *Sensors & Actuators B: Chemical*, 242: 476-483.

McHutchion, L., Zhao, L., Dixon, S., & Seelaus, R. (2020). Thresholds of perceivable color difference in silicone: A pilot study. *Maxillofacial Prosthetics (iJMP)*, 12-22.

Mehta, S., Patel, A., & J. Mehta. (2015). CCD or CMOS Image sensor for photography. *2015 International Conference on Communications and Signal Processing (ICCSP)* (pp. 291-294). IEEE.

Middleton Spectral Vision. (2021). Lumo Scanner for Specim Spectral Imaging. *https://www.middletonspectral.com/product/lumo-scanner-specim/*.

Ministry for Primary Industries. (2021). *Situation and outlook for primary industries.* Wellington, New Zealand: Ministry for Primary Inudstries, Economic Intelligence Unit.

Mishra, M., & Srivastava, M. (2014). A view of artificial neural networks. *IEEE International Conference on Advances in Engineering & Technology Research (ICAETR - 2014)* (pp. 1-3). Unnao, India: 2014.

Mulcare, D. C., & Coward, T. J. (2019). Suitability of a mobile phone colorimeter application for use as an objective aid when matching skin color during the fabrication of a maxillofacial prosthesis. *Prosthrodontics*, 28, 934-943.

Mutlu, A. Y., Kilic, V., Ozdemir, G. K., Bayram, A., Horzum, N., & Solmaz, M. E. (2017). Smartphone Based Colorimetric Detection via Machine Learning. *Analyst*, 13(142), 2434-2441.

New Zealand Kiwifruit Growers Incorporated. (2019, August 08). *The kiwifruit regulatory model*. Retrieved from New Zealand Kiwifruit Growers Incorporated.: https://www.nzkgi.org.nz/the-kiwifruit-regulatory-model/

New Zealand Kiwifruit Growers Incorporated. (2020). *Zespri annual 2019/20 key performance indicators.* Tauranga, New Zealand: New Zealand Kiwifruit Growers Incorporated.

Nguyen, R. M., Prasad, D. K., & Brown, M. S. (2014). Training-Based Spectral Reconstructionfrom a Single RGB Image. *Computer Vision -- ECCV 2014* (pp. 186-201). Cham, Switzerland: Springer International Publishing.

Nitze, I., Schulthess, U., & Asche, H. (2021). Comparison of machine learning algorithms random forest, artificial neural netowork and support vector machine to maximum likelihood for supervised crop type classification. *Proceedings of the 4th GEOBIA* (pp. 35-40). Rio de Janeiro, Brazil: Brazilian National Institute for Space Research (INPE),.

Nixon, M., Outlaw, F., MacDonald, L. W., & Leung, T. S. (2019). The importance of a device specific calibration for smartphone colorimetry. *27th Color and Imaging Conference.*

Norwegian Colour and Visual Computing Laboratory. (2021). QuickEval. *https://www.quickeval.no/*.

Ohno, Y. (2007). Spectral color measurements. In J. Schanda, *Colorimetry: understanding the CIE system* (pp. 101-134). Hoboken, New Jersey, USA: John Wiley & Sons.

Ohta, N., & Robertson, A. R. (2005). *Colorimetry fundamentals and applications.* West Sussex, England: John Wiley & Sons, Ltd.

Olkkonen, M., & Ekroll, V. (2016). Color constancy and contextual effects on colour apperance. In J. Kremers, R. C. Baraas, & N. J. Marshall, *Human color vision* (pp. 159-188). Cham, Switzerland: Springer International Publishing.

O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint*, arXiv:1511.08458.

Outlaw, F., Nixon, M., Odeyemi, O., MacDonald, L. W., Meek, J., & Leung, T. S. (2020). Smartphone screening for neonatal jaundice via ambient-subtracted sclera chromaticity. (pp. 15(3), e0216970). PLoS ONE.

Park, S. M., Visbal-Onufrak, M. A., Haque, M., Were, M. C., Naanyu, V., Hasank, K., & Kim, Y. L. (2020). mHealth spectroscopy of blood hemoglobin with spectral super-resolution. *Optica*, 20(7), 563-573.

Parkkinen, J. P., Hallikainen, J., & Jasskelinen, T. (1989). Characteristic spectra of munsell colors. *JOSA*, A 6(2), 318-322.

Parulski, K., & Spaulding, K. (2003). Color image processing for digital cameras. In G. Sharma, *Digital color imaging handbook* (pp. 728-757). Boca Raton, Florida, USA: CRC Press LLC.

Prabha, D. S., & Kumar, J. S. (2015). Assessment of banana fruit maturity by image processingtechnique. *Food Sci Technol*, 52(3): 1316-1327.

Qin, J., & Lu, R. (2008). Measurement of the optical properties of fruits and vegetables using spatially. *Postharvest Biology & Technology*, 49, 355-365.

Racek, F., & Baláž, T. (2010). Spectral Angle Mapper as a Tool for Matchingthe Spectra in Hyperspectral Processing. *Advances in Military Technology*, 7(2): 65-76.

Radio New Zealand. (2021, April 8). *Managing China's theft of NZ's SunGold kiwifruit*. Retrieved from RNZ: https://www.rnz.co.nz/programmes/the-detail/story/2018790587/managing-china-s-theft-of-nz-s-sungold-kiwifruit

Samsung. (2017). Samsung Galaxy S8. *https://www.samsung.com/nz/smartphones/galaxy-s8/*.

Sappi Fine Paper North America. (2013). *Defining and communicating color: the CIELAB system.* Sappi Fine Paper North America,.

Sathyanarayana, S. (2014). A gentle introduction to backpropagation. *Numeric Insight 7*, 1-15.

Schaare, P. N., & Fraser, D. G. (2000). Comparison of reflectance, interactance and transmissionmodes of visible-near infrared spectroscopy for measuringinternal properties of kiwifruit (Actinidia chinensis). *Postharvest Biology and Technology* , 20: 175-184.

Shen, L., Hagen, J. A., & Papautsky, I. (2012). Point-of-care colorimetric detection with a smartphone. *Lab on a Chip*, 12, 4240-4243.

Shengxian, C., Bangkui, D., Jiawei, S., Fan, L., Shanrang, Y., & Zhiming, X. (2010). A Colour Constancy Algorithm Based on Neural Network and Application. *Proceedings of the 8th World Congress on Intelligent Control and Automation* (pp. 3100-3103). Jinan, China: IEEE.

Shi, L., & Funt, B. (2000, Sept 18). *Re-processed version of the gehler color constancy dataset of 568 images*. Retrieved from http://www.cs.sfu.ca/~colour/data/

Shi, W., Loy, C. C., & Tang, X. (2016). Deep specialized network for illuminant estimation. *The 14th European Conference on Computer Vision (ECCV).* Amsterdam, The Netherlands: IEEE.

Shi, Z., Chen, C., Xiong, Z., Liu, D., & F. Wu. (2018). HSCNN+: Advanced CNN-Based Hyperspectral Recovery from RGB Images. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (pp. 1052-1058). IEEE.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint*, arXiv:1409.1556.

Specim Spectral Imaging Ltd. (2021). Specim FX10. *https://www.specim.fi/products/specim-fx10/*.

Stangierski, J., Weiss, D., & Kaczmarek, A. (2019). Multiple regression models and Artificial Neural Network (ANN) as prediction tools of changes in overall quality during the storage of spreadable processed Gouda cheese. *European Food Research and Technology*, 245: 2539-2547.

Stanikunas, R., Vaitkevicius, H., & Kulikowski, J. J. (2004). Investigation of color constancy with a neural network. *Neural Networks*, 17: 327-337.

Statista. (2022, Feb 5). *Production volume of kiwis worldwide in 2020, by leading country*. Retrieved from Statista: https://www.statista.com/statistics/812434/production-volume-of-leading-kiwi-producing-countries/

Stiebel, T., & Merhof, D. (2020). Brightness invariant deep spectral super-resolution. *Sensors*, 20, 5789.

Stiebel, T., Koppers, S., Seltsam, P., & Merhof, D. (2018). Reconstructing spectral images from RGB-images using a convolutional neural network. *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (pp. 1061-1066). IEEE.

Sulistyo, S. B., Woo, W. L., & Dlay, S. .. (2016). Computational intelligent color normalization for wheat plant images to support precision farming. *8th International Conference on Advanced Computational Intelligence* (pp. 130-135). Chiang Mai, Thailand: IEEE.

Susstrunk, S., Holm, J., & Finlayson, G. D. (2002). Chromatic adaption performance of different RGB sensors. *Proc. SPIE 4300, Color Imaging: Device-Independent Color, Color Hardcopy, and Graphic Arts VI.*

Szipszky, C., Aartsen, D. V., Criddle, S., Rao, P., Zentner, I., Justine, M., . . . Heysell, S. K. (2021). Determination of Rifampin Concentrations by Urine Colorimetry and Mobile Phone Readout for Personalized Dosing in Tuberculosis Treatment. *Pediatric Infectious Diseases Society*, 10(2), 104-111.

Taneja, P., Vasava, H. K., Daggupati, P., & Biswas, A. (2021). Multi-algorithm comparison to predict soil organic matter and soil moisture content from cell phone images. *Geoderma*, 285: 114863.

tania, M. H., Lwin, K. T., Shabut, A. M., Najlah, M., Chim, J., & Hossain, M. A. (2020). Intelligent image-based colourimetric tests using machione learning framework for lateral flow assays. *Expert Sytems With Applications*, 139: 112843.

The BabelColor Company. (2022, Jan 10). *The ColorChecker Pages (2/3).* Retrieved from BabelColor: https://babelcolor.com/colorchecker-2.htm#CCP2_data

The MathWorks Inc. (2021, June). *Smoothdata*. Retrieved from MathWorks: https://au.mathworks.com/help/matlab/ref/smoothdata.html

The MathWorks Inc. (2022, Feb 6). *fitrsvm*. Retrieved from MathWorks: https://au.mathworks.com/help/stats/fitrsvm.html?s_tid=doc_ta

The MathWorks Inc. (2022, Feb 17). *Trapz*. Retrieved from MathWorks: https://au.mathworks.com/help/matlab/ref/trapz.html

Tominaga, S., Nishi, S., & Ohtera, R. (2021). Measurement and estimation of spectral sensitivity functions for mobile phone cameras. *Sensors*, 21, 4985.

Tripathy, S., Reddy, M. S., Vanjari, S. R., Jana, S., & Singh, S. G. (2019). A step towards miniaturized milk adulteration detection system: smartphone-based accurate pH sensing using electrospun halochromix nanofribers. *Food Analytical Methods*, 12: 612-624.

Tsukida, K., & Gupta, M. R. (2011). *How to Analyze Paired Comparison Data.* Fort Belvoir, VA, USA: Defense Technical Information Center.

Verhoven, G. J. (2010). It's all about the format - unleashing the power of RAW aerial photography. *International Journal of Remote Sensing*, 31(8), 2009-2042.

Vienot, F., & Valraven, P. (2007). Color-matching functions: pyschologyical basis. In J. Schanda, *Colorimetry: understanding the CIE system* (pp. 219-240). Hoboken, New Jersey, USA: John Wiley & Sons, Inc.

Wang, X., & Zhang, D. (2013). A new tongue colorchecker design by space representation for precise correction. *Biomedical and Health Infomatics*, 17(2) 2168-2194.

Wang, X., & Zhang, D. (2013). A new tongue colorchecker design by space representation for precise correction. *Biomedical and Health Infomatics*, 17, 381-391.

Weijer, J. v., & Garrote, E. (2018). Adversarial Networks for spatial context-aware spectral image reconstruction from RGB. *arXiv*, 1709.00265v2.

Wikipedia. (2022, Feb 17). *Opponent Process*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Opponent_process

Wu, J. (2017). Introduction to convolutional nerual networks. *National Key Lab for Novel Software Technology*.

Xiong, Z., Shi, Z., Li, H., Wang, L., Liu, D., & Wu, F. (2017). HSCNN: CNN-Based Hyperspectral Image Recovery from Spectrally Undersampled Projections. *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, 518-525.

X-Rite Incorporated. (2009). *X-Rite.* Retrieved from ColorChecker: https://xritephoto.com/documents/literature/en/ColorData-1p_EN.pdf

Yan, Y., Zhang, L., Li, J., Wei, W., & Zhang, Y. (2018). Accurate spectral super- resolution from single RGB image using multi-scale CNN. *arXiv*, 1806.03575v3.

Yang, J., & Xu, Y. (2021). Prediction of fruit quality based on the RGB values of time-temperature indicator. *Food Science*, 86(3): 932.

Yuan, T., & Li, X. (2019). Fully convolutional color constancy with adding pooling. *11th International Conference on Communication Software and Networks* (pp. 666-671). IEEE.

Zapryanov, G., Ivanova, D., & Nikolova, I. (2012). Automatic white balance algorithms for digital still cameras - a comparative study. *Information technologies and control*, 1, 16-22.

Zespri Global Extension Team. (2021a). *Grower information pack, harvest 2021.* Tauranga: Zespri Group Ltd.

Zespri Group Limited. (2022b, Jan 8). *Zespri system assurance programes*. Retrieved from Zespri Kiwifruit: https://www.zespri.com/en-NZ/zespri-system-assurance-program

Zespri Group Ltd. (2019, December 04). *Zespri to commercialise red kiwifruit*. Retrieved from Zespri Kiwifruit: https://www.zespri.com/en-NZ/newsroomdetail/red-kiwifruit

Zespri Group Ltd. (2021b, May 27). *Zespri marks record season 2020/21*. Retrieved from Zespri Kiwifruit: https://www.zespri.com/en-NZ/newsroomdetail/2020-21-season-results

Zespri Group Ltd. (2022). *Zespri Quality Manual Feburary 2022.* Tauranga, NZ: Zespri Qaulity Assurance Team.

Zespri Group Ltd. (2022a). *Zespri Quality Manual Feburary 2022.* Tauranga, NZ: Zespri Qaulity Assurance Team.

Zhang, Y., Wu, Y., Zhang, Y., & Ozcan, A. (2016). Color calibration and fusion of lens-free and mobile-phone microscopy images for high-resolution and accurate color reproduction. *Scientific Reports*, 6:27811.

Zhao, J., Kechasov, D., Rewald, B., Bodner, G., Verheul, M., Clarke, N., & Clarke, J. L. (2020). Deep learning in hyperspectral image reconstruction from single RBG images - a case study on tomato quailty features. *Remote Sensing*, 12: 3258.

Zhao, Y., Zhou, C. E., & Rafferty, K. (2018). Spectral illumination correction" achieving relative color constancy under the spectral domain.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2015). Learning Deep Features for Discriminative Localization. *arXiv*, 1512.04150.

Zhou, R., Wu, D., Zhou, R., Fang, L., Zheng, X., & Lou, X. (2019). Estimation of DBH at forest stand level based on multi-parameters and generalized regression neural network. *Forests*, 10: 788.

Zhou, X., Wang, P., Dai, G., Yan, J., & Yang, Z. (2021). Tomato fruit maturity detection method based on YOLOV4 and statistical color model. *Proceedings of 11th IEEE International Conferenceon CYBER Technology in Automation, Control, and Intelligent Systems* (pp. 904-908). Jiaxing, China: IEEE.

8.0 Appendix

8.1 Network Errors of GRNN

8.1.2 Error over Visible Range

Table 8.1: Table of Errors from GRNN over the Visible Range

Network	Test RMSE	Mean Dist	Med Dist	Std Dist	Effc Fail Rate
Whole	282.66	224.10	174.89	178.82	0.1017
Core	1633.51	1341.47	1229.63	935.08	0.5932
Inner	704.30	580.69	539.56	401.46	0.2429
Outer	266.80	206.04	173.33	171.21	0.1299
Whole + Core	407.94	304.80	237.37	276.43	0.1073
Whole + Inner	407.49	208.20	145.81	466.76	0.0904
Whole + Outer	268.17	216.97	191.96	157.42	0.1073
Core + Inner	716.34	569.27	473.79	443.04	0.2655
Core + Outer	326.04	234.02	179.89	238.90	0.0847
Inner + Outer	253.35	195.73	146.27	159.52	0.0621
Whole + Core + Inner	272.35	202.68	131.89	178.51	0.0678
Whole + Core + Outer	209.97	171.78	141.18	124.48	0.0847
Whole + Inner + Outer	222.02	179.63	150.20	132.29	0.0904
Core + Inner + Outer	304.23	203.42	167.32	173.08	0.0791
Whole + core + inner + outer	281..70	220.88	189.24	176.22	0.1130

8.1.2 Error over Visible NIR Range

Table 8.2: Table of Errors from GRNN over the Visible NIR Range

Network	Test RMSE	Mean Dist	Med Dist	Std Dist	Effc Fail Rate
Whole	496.54	269.62	181.27	539.55	0.1017
Core	1612.76	1249.67	1088.27	1024.13	0.5367
Inner	826.38	689.59	618.35	465.17	0.3164
Outer	286.75	207.69	162.34	210.62	0.1130
Whole + Core	265.79	214.91	185.72	166.93	0.1017
Whole + Inner	245.21	189.63	156.24	158.04	0.0960
Whole + Outer	289.16	222.47	176.60	176.91	0.0904
Core + Inner	895.05	558.92	361.11	730.48	0.2881
Core + Outer	334.15	214.47	159.11	261.23	0.1186
Inner + Outer	234.59	184.68	150.89	153.68	0.1073
Whole + Core + Inner	423.22	339.60	307.95	249.43	0.1695
Whole + Core + Outer	223.06	172.95	129.92	145.73	0.0678
Whole + Inner + Outer	297.30	190.52	160.47	227.28	0.0904

Core + Inner + Outer	517.12	328.28	275.30	432.98	0.1243
Whole + core + inner + outer	243.48	196.71	164.07	149.63	0.0960

8.2 Network Error of Illuminant Invariant CNN

8.2.1 Error over Visible Range

Table 8.3: Table of Errors from CNN over the Visible Range

VISIBLE RANGE	Count	Integral Error	MRAE	RMSE	SAM
Whole Network **Std**	1221	0.6774 0.3763	0.0147 0.0098	0.0057 0.0032	0.0174 0.0069
Grower – 48785	247	0.8200	0.0182	0.0069	0.0194
Grower – 27844	257	0.7238	0.0153	0.0061	0.0174
Grower – 7829e	255	1.0803	0.0244	0.0094	0.0225
Grower – 5205a	205	0.8537	0.0192	0.0071	0.0173
Grower – 98948	36	0.8181	0.0198	0.0070	0.0169
Grower – 9205j	36	0.7387	0.0172	0.0060	0.0171
Grower – 7174a	91	0.6918	0.0152	0.0055	0.0161
Grower – 87143	29	0.5677	0.0171	0.0051	0.0110
Grower – 77357	6	0.8364	0.0208	0.0074	0.0183
Grower – 32204	26	0.5634	0.0120	0.0047	0.0154
Grower – 26278	33	0.6774	0.0147	0.0057	0.0174
Size – 25	69	0.6529	0.0138	0.0055	0.0159
Size – 27	35	0.7935	0.0157	0.0066	0.0185
Size – 30	81	0.8896	0.0190	0.0075	0.0201
Size – 33	124	0.7611	0.0173	0.0066	0.0173
Size – 36	322	0.7341	0.0172	0.0063	0.0164
Size – 39	109	0.8124	0.0174	0.0067	0.0205
Size – 42	229	0.8636	0.0198	0.0073	0.0192
Size – 46	70	1.1002	0.0225	0.0092	0.0249
Size – 49	166	0.7481	0.0156	0.0062	0.0191
Size – NA	49	0.7517	0.0174	0.0062	0.0188
iPhone8	123	0.8550	0.0201	0.0073	0.0189
iPhone11	122	0.7668	0.0171	0.0065	0.0177
iPhone12max	122	0.7948	0.0177	0.0067	0.0186
p10lite	134	0.7317	0.0159	0.0062	0.0181
GalaxyNote10Pro	119	0.8190	0.0179	0.0069	0.0192
Galaxys20	103	0.8018	0.0175	0.0067	0.0184
Aquos5G	125	0.8265	0.0182	0.0070	0.0197
RedMiNote9s	131	0.8167	0.0183	0.0069	0.0182
SonyExperia5-2	119	0.8077	0.0179	0.0068	0.0187
Pixel4	123	0.7732	0.0170	0.0066	0.0179

8.2.2 Error over the visible NIR range

Table 8.4: Table of Errors from CNN over the Visible NIR Range

NIR RANGE	Count	Integral Error	MRAE	RMSE	SAM
Whole Network **Std**	1221	1.0783 0.4237	0.0117 0.0051	0.0062 0.0024	0.0151 0.0061
Grower – 48785	240	0.9376	0.0098	0.0054	0.0140
Grower – 27844	258	1.0983	0.0126	0.0061	0.0135
Grower – 7829e	218	0.9862	0.0110	0.0054	0.0116
Grower – 5205a	206	1.2569	0.0133	0.0072	0.0178
Grower – 98948	38	1.1518	0.0116	0.0069	0.0180
Grower – 9205j	37	1.1653	0.0115	0.0070	0.0191
Grower – 7174a	96	1.1175	0.0123	0.0070	0.0177
Grower – 87143	37	0.9229	0.0099	0.0053	0.0148
Grower – 77357	14	0.5903	0.0057	0.0038	0.0109
Grower – 32204	28	1.6438	0.0172	0.0099	0.0274
Grower – 26278	49	1.0649	0.0125	0.0062	0.0161
Size – 25	59	1.0481	0.0111	0.0060	0.0152
Size – 27	42	0.9654	0.0104	0.0056	0.0138
Size – 30	86	1.1235	0.0113	0.0062	0.0156
Size – 33	119	1.0696	0.0112	0.0061	0.0152
Size – 36	339	1.0826	0.0115	0.0064	0.0167
Size – 39	122	0.9901	0.0109	0.0056	0.0143
Size – 42	217	1.0743	0.0119	0.0060	0.0141
Size – 46	70	1.1462	0.0126	0.0064	0.0147
Size – 49	109	1.1011	0.0129	0.0060	0.0123
Size – NA	58	1.1597	0.0132	0.0072	0.0166
iPhone8	139	1.3156	0.0143	0.0075	0.0185
iPhone11	110	1.0906	0.0121	0.0062	0.0149
iPhone12max	114	1.0272	0.0113	0.0060	0.0149
p10lite	123	1.0626	0.0116	0.0061	0.0148
GalaxyNote10Pro	117	1.0551	0.0116	0.0060	0.0144
Galaxys20	127	1.0295	0.0111	0.0058	0.0143
Aquos5G	121	1.0465	0.0115	0.0060	0.0145
RedMiNote9s	125	1.1402	0.0120	0.0064	0.0157
SonyExperia5-2	122	1.0164	0.0111	0.0059	0.0146
Pixel4	123	0.9881	0.0107	0.0056	0.0138

8.3 Network Error of Combined Dataset

8.3.1 Error over the Visible Range

Table 8.5: Table of Errors from CNN over the Visible Range using the combined dataset

VISIBLE RANGE	Count	Integral Error	MRAE	RMSE	SAM
Whole Network **Std**		1.0520 0.7486	0.0259 0.0192	0.0091 0.0062	0.0223 0.0153
Real RAW	1636	1.4265	0.0355	0.0123	0.0287
Synthetic RAW	1226	0.5522	0.0131	0.0048	0.0136
Grower – 48785	439	1.07583	0.02629	0.00926	0.02526
Grower – 27844	544	0.95866	0.02359	0.00832	0.02135
Grower – 7829e	464	0.92132	0.02209	0.00795	0.01831
Grower – 5205a	348	0.99668	0.02390	0.00857	0.01936
Grower – 98948	206	1.03161	0.02477	0.00890	0.02120
Grower – 9205j	34	0.56975	0.01353	0.00493	0.01398
Grower – 7174a	204	1.11511	0.02897	0.00981	0.02444
Grower – 87143	208	1.27015	0.03254	0.01097	0.02646
Grower – 77357	149	1.33900	0.03322	0.01164	0.02757
Grower – 32204	154	1.42203	0.03498	0.01225	0.02818
Grower – 26278	112	0.89763	0.02365	0.00774	0.01761
Size – 25	114	0.9094	0.0221	0.0078	0.0182
Size – 27	149	1.2263	0.0298	0.0105	0.0208
Size – 30	160	0.8955	0.0218	0.0077	0.0180
Size – 33	221	0.8931	0.0220	0.0079	0.0200
Size – 36	1201	1.1448	0.0287	0.0099	0.0236
Size – 39	213	0.8905	0.0218	0.0077	0.0204
Size – 42	401	0.9201	0.0220	0.0080	0.0214
Size – 46	204	1.5205	0.0380	0.0129	0.0328
Size – 49	136	0.7880	0.0184	0.0069	0.0172
Size – NA	63	0.5215	0.0116	0.0048	0.0154
Synth - iPhone8	120	0.6255	0.0151	0.0055	0.0154
Synth - iPhone11	128	0.5131	0.0122	0.0045	0.0127
Synth - iPhone12max	125	0.5540	0.0125	0.0047	0.0138
Synth - p10lite	122	0.4217	0.0099	0.0037	0.0108
Synth - GalaxyNote10Pro	125	0.6295	0.0139	0.0054	0.0164
Synth - Galaxys20	139	0.5145	0.0123	0.0045	0.0129
Synth - Aquos5G	107	0.4805	0.0114	0.0042	0.0110
Synth - RedMiNote9s	108	0.7280	0.0187	0.0065	0.0174
Synth - SonyExperia5-2	128	0.5464	0.0134	0.0048	0.0128
Synth - Pixel4	124	0.5272	0.0117	0.0045	0.0129
Real – iPhone12 Pro	863	1.3236	0.0332	0.0114	0.0272
Real – Galaxy S8	773	1.5414	0.0382	0.0133	0.0305

8.3.2 Error over the Visible Range

Table 8.6: Table of Errors from CNN over the Visible Range using the combined dataset

NIR RANGE	Count	Integral Error	MRAE	RMSE	SAM
Whole Network **Std**	2862	1.7935 1.1653	0.0222 0.0162	0.0096 0.0060	0.0175 0.0115
Real RAW	1638	2.3313	0.0292	0.0126	0.0233
Synthetic RAW	1224	1.0738	0.0128	0.0057	0.0097
Grower – 48785	483	1.6533	0.0202	0.0089	0.0164
Grower – 27844	540	1.6010	0.0194	0.0087	0.0165
Grower – 7829e	474	1.6142	0.0193	0.0087	0.0160
Grower – 5205a	335	1.8744	0.0228	0.0100	0.0174
Grower – 98948	205	1.7095	0.0203	0.0093	0.0174
Grower – 9205j	25	1.3335	0.0167	0.0066	0.0084
Grower – 7174a	202	2.0116	0.0275	0.0105	0.0159
Grower – 87143	206	2.1050	0.0265	0.0112	0.0210
Grower – 77357	137	2.0983	0.0256	0.0114	0.0219
Grower – 32204	167	2.4802	0.0329	0.0132	0.0248
Grower – 26278	88	1.7216	0.0225	0.0090	0.0136
Size – 25	102	1.5465	0.0193	0.0083	0.0147
Size – 27	149	1.7463	0.0203	0.0095	0.0177
Size – 30	175	1.5686	0.0192	0.0084	0.0138
Size – 33	199	1.5640	0.0186	0.0084	0.0158
Size – 36	1152	1.9875	0.0253	0.0106	0.0187
Size – 39	244	1.6379	0.0202	0.0088	0.0163
Size – 42	404	1.6384	0.0195	0.0089	0.0174
Size – 46	230	2.2082	0.0273	0.0119	0.0227
Size – 49	146	1.3733	0.0165	0.0073	0.0132
Size – NA	61	1.1445	0.0143	0.0060	0.0095
Synth - iPhone8	139	0.9739	0.0112	0.0054	0.0117
Synth - iPhone11	125	1.0771	0.0131	0.0056	0.0087
Synth - iPhone12max	120	1.1882	0.0139	0.0061	0.0097
Synth - p10lite	127	0.9770	0.0121	0.0051	0.0083
Synth - GalaxyNote10Pro	121	1.0067	0.0118	0.0052	0.0092
Synth - Galaxys20	116	1.0687	0.0128	0.0056	0.0093
Synth - Aquos5G	118	1.1745	0.0144	0.0061	0.0095
Synth - RedMiNote9s	125	1.0860	0.0125	0.0059	0.0117
Synth - SonyExperia5-2	109	1.1640	0.0141	0.0061	0.0098
Synth - Pixel4	124	1.0535	0.0128	0.0055	0.0093
Real – iPhone12 Pro	845	2.2607	0.0287	0.0122	0.0224
Real – Galaxy S8	793	2.4066	0.0298	0.0130	0.0242

8.3 Filter Visualisation

8.3.1 Filter of CNN Trained Independently from GRNN

Figure 8.1: Filters of KiwiNet from CNN layer when trained separately

8.3.2 Filter of CNN Trained with GRNN

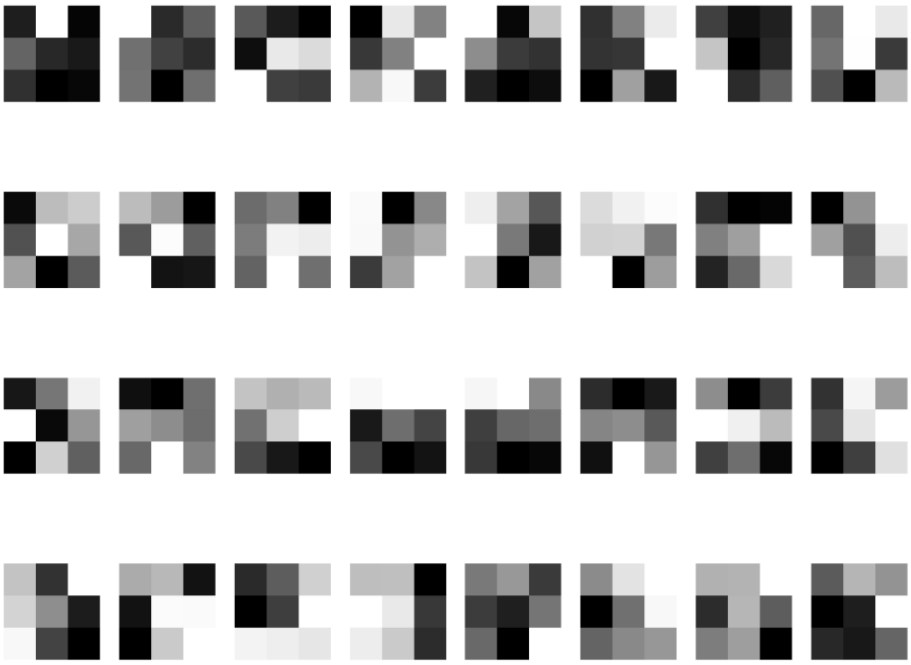


Figure 8.2: Filters of KiwiNet from CNN layer when trained together

8.4 Hyperparameter Sweep Configurations

8.4.1 Initial Sweep

```
method: random
metric:
  goal: minimize
  name: loss
parameters:
  VIS:
    values:
    - 0
    - 1
  batch_size:
    distribution: q_log_uniform
    max: 4.852030263919617
    min: 3.4657359027997265
    q: 1
  core:
    values:
    - 0
  dropout:
    values:
    - 0.1
    - 0.15
    - 0.2
    - 0.25
    - 0.3
  epochs:
    values:
    - 100
  inner:
```

```
  values:
  - 0
 learning_rate:
  distribution: uniform
  max: 0.01
  min: 0
 optimizer:
  values:
  - adam
  - sgd
 outer:
  values:
  - 0
  - 1
 whole:
  values:
  - 0
  - 1
```

8.4.2 Second Hyperparameter Sweep

```
method: random
metric:
 goal: minimize
 name: Train_loss
parameters:
 VIS:
  values:
  - 0
  - 1
 batch_size:
```

distribution: q_log_uniform
max: 4.852030263919617
min: 3.4657359027997265
q: 1
core:
values:
- 0
dropout:
values:
- 0.1
- 0.2
- 0.3
- 0.4
- 0.5
epochs:
values:
- 100
fc1:
values:
- 0.5
- 0.75
- 1
- 1.25
- 1.5
fc2:
values:
- 0.5
- 0.75
- 1
- 1.25

```
  - 1.5
fc3:
  values:
  - 1
inner:
  values:
  - 0
learning_rate:
  distribution: uniform
  max: 0.01
  min: 0
optimizer:
  values:
  - adam
outer:
  values:
  - 0
  - 1
whole:
  values:
  - 0
  - 1
```

8.5 Synthesising Raw Camera Data Matlab Code

8.5.1 Matlab Code Hyperspectral to Camera Raw, Part 1

```
clear all
close all


folder = 'D:\MASTERS\Sam Masters Data\filtered_ref';
save_loc = 'D:\MASTERS\CamRaw_Generated\dataset';
```

```
cd(folder)
files = dir;
filt_ref=files(~ismember({files.name},{'.','..'}));
nfiles = length(filt_ref);    % Number of files found

sensors_set1 =
{'iPhone8';'iPhone11';'iPhone12max';'p10lite';'GalaxyNote10Pro';'Galaxys20';
    'Aquos5G';'RedMiNote9s';'SonyExperia5-2';'Pixel4'};

for i=2:nfiles
    i
    for ph=1:length(sensors_set1)
        sensors_set1(ph);
        tic
        cd(folder)
        sensor = cell2mat(sensors_set1(ph));

        %grab random number from list
%         k = randoms(i);

        %load Hyperspecrtal image
        hsi = load(filt_ref(i).name);
        hsi = hsi.filt_reflectance;

        %set range of hyperspectral camera
        lambdaRef = [400 1000];

        [rawD50, lambdaRef,lambdaIll,lambdaObs] =
SamsRef2x2(hsi,'lambdaRef',lambdaRef,'ill','D50','sensor',sensor);
        [rawD65, lambdaRef,lambdaIll,lambdaObs] =
SamsRef2x2(hsi,'lambdaRef',lambdaRef,'ill','D65','sensor',sensor);
        [rawA, lambdaRef,lambdaIll,lambdaObs] =
SamsRef2x2(hsi,'lambdaRef',lambdaRef,'ill','A','sensor',sensor);
        [rawF2, lambdaRef,lambdaIll,lambdaObs] =
SamsRef2x2(hsi,'lambdaRef',lambdaRef,'ill','F2','sensor',sensor);
        [rawH, lambdaRef,lambdaIll,lambdaObs] =
SamsRef2x2(hsi,'lambdaRef',lambdaRef,'ill','HSI_halo','sensor',sensor);


%         figure(1)
%         subplot(3,2,1), imshow(rawD50)
%         title('D50')
%         subplot(3,2,2), imshow(rawD65)
%         title('D65')
%         subplot(3,2,3), imshow(rawA)
%         title('A')
%         subplot(3,2,4), imshow(rawF2)
%         title('F2')
%         subplot(3,2,5), imshow(rawH)
%         title('H')

        name = filt_ref(i).name(1:length(filt_ref(i).name)-4);

        phonename = cell2mat(sensors_set1(ph));
        d50name = strcat(name,'--',phonename,'_O_D50.png');
```

```
        d65name = strcat(name,'--',phonename,'_O_D65.png');
        Aname = strcat(name,'--',phonename,'_O_A.png');
        F2name = strcat(name,'--',phonename,'_O_F2.png');
        Hname = strcat(name,'--',phonename,'_O_H.png');

        cd(save_loc)
        imwrite(rawD50, d50name, 'BitDepth',16)
        imwrite(rawD65, d65name, 'BitDepth',16)
        imwrite(rawA, Aname, 'BitDepth',16)
        imwrite(rawF2, F2name, 'BitDepth',16)
        imwrite(rawH, Hname, 'BitDepth',16)
        toc
        pause

    end
    i
end
```

8.5.2 Matlab Code Hyperspectral to Camera Raw, Part 1

```
%% REF2X  Colorimetric rendering from reflectance data
%
%   EXAMPLES
%   - srgb = ref2x(ref);
%   - srgb = ref2x(ref, 'lambdaRef', 400:10:720);
%   - lab  = ref2x(ref, 'colorspace', 'lab');
%
%   INPUT
%   - ref: reflectance image, can be either a 2D matrix (pixels in rows,
bands in columns)
%     or a hyperspectral image.
%   - varargin: parameters (see below)
%
%   OUTPUT
%   - result: trichromatic image with the same spatial dimension(s) as INPUT.
%
%   PARAMETERS (optional)
%   - 'lambdaRef': wavelength of each channel in 'ref' (default: 400:10:700)
%   - 'ill': power distribution of the illuminant (default: d65)
%   - 'lambdaIll': wavelength of each channel in ill (default: 400:10:700)
%   - 'colorspace': output colorspace
%       'srgb' (default)
```

```
%       'xyz'
%       'xyY'
%       'lab'
%       'lab2000HL'
%       'linearLAB'
%   - 'sensor': type of standard sensor
%       '1931' for 2° CIE Standard Observer (default)
%       '1964' for 10° CIE Standard Observer
%       'camera1' for RGB camera model 1
%   - 'CAT': iCAM06 chromatic adaptation transform (not used by default)
%   - 'bitDepth' (under construction)

% contact: steven.lemoan@ntnu.no
% last update: July 2021 (version 2 beta)

function[result,lambdaRef,lambdaIll,lambdaObs] = SamsRef2x2(ref, varargin)

%% Parameters

% Default
colorspace = 'srgb';
lambdaRef  = [400 720];
obs        = '1931';
% ill        = 'D65';
CAT        = 0;

% Set from input
if (~isempty(varargin))
    for i = 1:2:size(varargin,2)
        if     (strcmp(varargin{i}, 'lambdaRef'))
            lambdaRef = varargin{i+1};
        elseif (strcmp(varargin{i}, 'ill'))
            ill = varargin{i+1};
        elseif (strcmp(varargin{i}, 'lambdaIll'))
            lambdaIll = varargin{i+1};
        elseif (strcmp(varargin{i}, 'sensor'))
            obs = varargin{i+1};
        elseif (strcmp(varargin{i}, 'CMF'))
            barsCUSTOM = varargin{i+1};
            obs = 'custom';
        elseif (strcmp(varargin{i}, 'lambdaCMF'))
            lambdaCMF = varargin{i+1};
        elseif (strcmp(varargin{i}, 'CAT'))
            CAT = varargin{i+1};
        elseif (strcmp(varargin{i}, 'colorspace'))
            colorspace = varargin{i+1};
        elseif (strcmp(varargin{i}, 'bitDepth'))
            bd = varargin{i+1};
        else
            warning('ref2x:vargin', ['Wrong option ''%s'''!']);
        end
    end
end
%% CIE standard observer data
```

```
barsCIE = [ 360.0000    0.0001    0.0000    0.0006    0.0000    0.0000
0.0000
            365.0000    0.0002    0.0000    0.0011    0.0000    0.0000
0.0000
            370.0000    0.0004    0.0000    0.0019    0.0000    0.0000
0.0000
            375.0000    0.0007    0.0000    0.0035    0.0000    0.0000
0.0001
            380.0000    0.0014    0.0000    0.0065    0.0002    0.0000
0.0007
            385.0000    0.0022    0.0001    0.0105    0.0007    0.0001
0.0029
            390.0000    0.0042    0.0001    0.0201    0.0024    0.0003
0.0105
            395.0000    0.0076    0.0002    0.0362    0.0072    0.0008
0.0323
            400.0000    0.0143    0.0004    0.0679    0.0191    0.0020
0.0860
            405.0000    0.0232    0.0006    0.1102    0.0434    0.0045
0.1971
            410.0000    0.0435    0.0012    0.2074    0.0847    0.0088
0.3894
            415.0000    0.0776    0.0022    0.3713    0.1406    0.0145
0.6568
            420.0000    0.1344    0.0040    0.6456    0.2045    0.0214
0.9725
            425.0000    0.2148    0.0073    1.0391    0.2647    0.0295
1.2825
            430.0000    0.2839    0.0116    1.3856    0.3147    0.0387
1.5535
            435.0000    0.3285    0.0168    1.6230    0.3577    0.0496
1.7985
            440.0000    0.3483    0.0230    1.7471    0.3837    0.0621
1.9673
            445.0000    0.3481    0.0298    1.7826    0.3867    0.0747
2.0273
            450.0000    0.3362    0.0380    1.7721    0.3707    0.0895
1.9948
            455.0000    0.3187    0.0480    1.7441    0.3430    0.1063
1.9007
            460.0000    0.2908    0.0600    1.6692    0.3023    0.1282
1.7454
            465.0000    0.2511    0.0739    1.5281    0.2541    0.1528
1.5549
            470.0000    0.1954    0.0910    1.2876    0.1956    0.1852
1.3176
            475.0000    0.1421    0.1126    1.0419    0.1323    0.2199
1.0302
            480.0000    0.0956    0.1390    0.8130    0.0805    0.2536
0.7721
            485.0000    0.0580    0.1693    0.6162    0.0411    0.2977
0.5706
            490.0000    0.0320    0.2080    0.4652    0.0162    0.3391
0.4153
            495.0000    0.0147    0.2586    0.3533    0.0051    0.3954
0.3024
```

500.0000	0.0049	0.3230	0.2720	0.0038	0.4608
0.2185					
505.0000	0.0024	0.4073	0.2123	0.0154	0.5314
0.1592					
510.0000	0.0093	0.5030	0.1582	0.0375	0.6067
0.1120					
515.0000	0.0291	0.6082	0.1117	0.0714	0.6857
0.0822					
520.0000	0.0633	0.7100	0.0782	0.1177	0.7618
0.0607					
525.0000	0.1096	0.7932	0.0573	0.1730	0.8233
0.0430					
530.0000	0.1655	0.8620	0.0422	0.2365	0.8752
0.0305					
535.0000	0.2257	0.9149	0.0298	0.3042	0.9238
0.0206					
540.0000	0.2904	0.9540	0.0203	0.3768	0.9620
0.0137					
545.0000	0.3597	0.9803	0.0134	0.4516	0.9822
0.0079					
550.0000	0.4334	0.9950	0.0087	0.5298	0.9918
0.0040					
555.0000	0.5121	1.0000	0.0057	0.6161	0.9991
0.0011					
560.0000	0.5945	0.9950	0.0039	0.7052	0.9973
0					
565.0000	0.6784	0.9786	0.0027	0.7938	0.9824
0					
570.0000	0.7621	0.9520	0.0021	0.8787	0.9556
0					
575.0000	0.8425	0.9154	0.0018	0.9512	0.9152
0					
580.0000	0.9163	0.8700	0.0017	1.0142	0.8689
0					
585.0000	0.9786	0.8163	0.0014	1.0743	0.8256
0					
590.0000	1.0263	0.7570	0.0011	1.1185	0.7774
0					
595.0000	1.0567	0.6949	0.0010	1.1343	0.7204
0					
600.0000	1.0622	0.6310	0.0008	1.1240	0.6583
0					
605.0000	1.0456	0.5668	0.0006	1.0891	0.5939
0					
610.0000	1.0026	0.5030	0.0003	1.0305	0.5280
0					
615.0000	0.9384	0.4412	0.0002	0.9507	0.4618
0					
620.0000	0.8544	0.3810	0.0002	0.8563	0.3981
0					
625.0000	0.7514	0.3210	0.0001	0.7549	0.3396
0					
630.0000	0.6424	0.2650	0.0001	0.6475	0.2835
0					
635.0000	0.5419	0.2170	0.0000	0.5351	0.2283
0					

	640.0000	0.4479	0.1750	0.0000	0.4316	0.1798
0						
	645.0000	0.3608	0.1382	0.0000	0.3437	0.1402
0						
	650.0000	0.2835	0.1070	0	0.2683	0.1076
0						
	655.0000	0.2187	0.0816	0	0.2043	0.0812
0						
	660.0000	0.1649	0.0610	0	0.1526	0.0603
0						
	665.0000	0.1212	0.0446	0	0.1122	0.0441
0						
	670.0000	0.0874	0.0320	0	0.0813	0.0318
0						
	675.0000	0.0636	0.0232	0	0.0579	0.0226
0						
	680.0000	0.0468	0.0170	0	0.0409	0.0159
0						
	685.0000	0.0329	0.0119	0	0.0286	0.0111
0						
	690.0000	0.0227	0.0082	0	0.0199	0.0077
0						
	695.0000	0.0158	0.0057	0	0.0138	0.0054
0						
	700.0000	0.0114	0.0041	0	0.0096	0.0037
0						
	705.0000	0.0081	0.0029	0	0.0066	0.0026
0						
	710.0000	0.0058	0.0021	0	0.0046	0.0018
0						
	715.0000	0.0041	0.0015	0	0.0031	0.0012
0						
	720.0000	0.0029	0.0010	0	0.0022	0.0008
0						
	725.0000	0.0020	0.0007	0	0.0015	0.0006
0						
	730.0000	0.0014	0.0005	0	0.0010	0.0004
0						
	735.0000	0.0010	0.0004	0	0.0007	0.0003
0						
	740.0000	0.0007	0.0002	0	0.0005	0.0002
0						
	745.0000	0.0005	0.0002	0	0.0004	0.0001
0						
	750.0000	0.0003	0.0001	0	0.0003	0.0001
0						
	755.0000	0.0002	0.0001	0	0.0002	0.0001
0						
	760.0000	0.0002	0.0001	0	0.0001	0.0000
0						
	765.0000	0.0001	0.0000	0	0.0001	0.0000
0						
	770.0000	0.0001	0.0000	0	0.0001	0.0000
0						
	775.0000	0.0001	0.0000	0	0.0000	0.0000
0						

```
            780.0000    0.0000    0.0000         0    0.0000    0.0000
0
            785.0000    0.0000    0.0000         0    0.0000    0.0000
0
            790.0000    0.0000    0.0000         0    0.0000    0.0000
0
            795.0000    0.0000    0.0000         0    0.0000    0.0000
0
            800.0000    0.0000    0.0000         0    0.0000    0.0000
0
            805.0000    0.0000    0.0000         0    0.0000    0.0000
0
            810.0000    0.0000    0.0000         0    0.0000    0.0000
0
            815.0000    0.0000    0.0000         0    0.0000    0.0000
0
            820.0000    0.0000    0.0000         0    0.0000    0.0000
0
            825.0000    0.0000    0.0000         0    0.0000    0.0000
0
            830.0000    0.0000    0.0000         0    0.0000    0.0000
0];

%% Formatting (1)

% reshaping ---------------------------------------------------------------
if ndims(ref)==3
    dims = size(ref);
    ref = reshape(ref,[dims(1)*dims(2) dims(3)]);
end
nbsamples = size(ref,1);

% conversion to double ----------------------------------------------------
if isa(ref,'uint8')
    ref = double(ref)./255;
end

% clipping ----------------------------------------------------------------
ref = min(max(ref,0),1);

% scaling -----------------------------------------------------------------
% ref = ref./max(ref(:)); % (for uncalibrated input only)

% computing wavelength range for reflectance ------------------------------
if length(lambdaRef)==2
    lambdaRef = lambdaRef(1):((lambdaRef(end)-lambdaRef(1))/(size(ref,2)-
1)):lambdaRef(end);
end

% truncating wavelength range to visible only -----------------------------
if strcmp(obs,'1931') || strcmp(obs,'1964') || strcmp(obs,'camera1') ||
strcmp(obs,'camera2')
    idx = lambdaRef>800;
    lambdaRef(idx) = [];
    ref(:,idx) = [];
```

```
end

%% Loading illuminant
if strcmp(ill,'D50')
    ill = 1;
elseif strcmp(ill,'D65')
    ill = 2;
elseif strcmp(ill,'HSI_halo')
    ill = 3;
elseif strcmp(ill,'A')
    ill = 5;
elseif strcmp(ill,'F2')
    ill = 6;
elseif strcmp(ill,'CC_tile')
    ill = 7;
end

[ill, lambdaIll] = readill(ill,lambdaRef(1),lambdaRef(end),5);

%                  INTERPOLATE RANGE DATA    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

x = ill;
y = lambdaIll;


xq = min(y):1:max(y);
interIll = interp1(y,x,xq);

ill = interIll;
lambdaIll = xq;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ill = ill(:)';
lambdaIll = lambdaIll(:)';

%% Loading observer or sensor

if strcmp(obs,'1931')
    lambdaObs = barsCIE(:,1);
    sensor = barsCIE(:,2:4); % CIE 1931

elseif strcmp(obs,'1964')
    lambdaObs = barsCIE(:,1);
    sensor = barsCIE(:,5:7); % CIE 1964

elseif strcmp(obs,'camera1')
    load Camera_responses
    lambdaObs = 400:10:700;
    sensor = responses(:,1:3);
    responses = responses';
    colorspace = 'raw';

elseif strcmp(obs,'camera2')
    load Camera_responses
    lambdaObs = 400:10:700;
```

```
    responses = responses';
    sensor = responses(:,4:6);
    colorspace = 'raw';

elseif strcmp(obs,'iPhone6')
    load('G:\My Drive\Zespri\MATLAB3\phone_sense.mat')
    lambdaObs = 400:10:700;
    sensor = cams(:,1:3);
    colorspace = 'raw';

elseif strcmp(obs,'iPhoneSE')
    load('G:\My Drive\Zespri\MATLAB3\phone_sense.mat')
    lambdaObs = 400:10:700;
    sensor = cams(:,4:6);
    colorspace = 'raw';

elseif strcmp(obs,'iPhone8')
    load('G:\My Drive\Zespri\MATLAB3\phone_sense.mat')
    lambdaObs = 400:10:700;
    sensor = cams(:,7:9);
    colorspace = 'raw';

elseif strcmp(obs,'iPhoneX')
    load('G:\My Drive\Zespri\MATLAB3\phone_sense.mat')
    lambdaObs = 400:10:700;
    sensor = cams(:,10:12);
    colorspace = 'raw';

elseif strcmp(obs,'iPhone11')
    load('G:\My Drive\Zespri\MATLAB3\phone_sense.mat')
    lambdaObs = 400:10:700;
    sensor = cams(:,13:15);
    colorspace = 'raw';

elseif strcmp(obs,'iPhone12max')
    load('G:\My Drive\Zespri\MATLAB3\phone_sense.mat')
    lambdaObs = 400:10:700;
    sensor = cams(:,16:18);
    colorspace = 'raw';

elseif strcmp(obs,'p10lite')
    load('G:\My Drive\Zespri\MATLAB3\phone_sense.mat')
    lambdaObs = 400:10:700;
    sensor = cams(:,19:21);
    colorspace = 'raw';

elseif strcmp(obs,'NovaLite2')
    load('G:\My Drive\Zespri\MATLAB3\phone_sense.mat')
    lambdaObs = 400:10:700;
    sensor = cams(:,22:24);
    colorspace = 'raw';

elseif strcmp(obs,'Galaxy7Edge')
    load('G:\My Drive\Zespri\MATLAB3\phone_sense.mat')
    lambdaObs = 400:10:700;
```

```
    sensor = cams(:,25:27);
    colorspace = 'raw';

elseif strcmp(obs,'Galaxy9')
    load('G:\My Drive\Zespri\MATLAB3\phone_sense.mat')
    lambdaObs = 400:10:700;
    sensor = cams(:,28:30);
    colorspace = 'raw';

elseif strcmp(obs,'GalaxyNote10Pro')
    load('G:\My Drive\Zespri\MATLAB3\phone_sense.mat')
    lambdaObs = 400:10:700;
    sensor = cams(:,31:33);
    colorspace = 'raw';

elseif strcmp(obs,'Galaxys20')
    load('G:\My Drive\Zespri\MATLAB3\phone_sense.mat')
    lambdaObs = 400:10:700;
    sensor = cams(:,34:36);
    colorspace = 'raw';

elseif strcmp(obs,'AquosS3lite')
    load('G:\My Drive\Zespri\MATLAB3\phone_sense.mat')
    lambdaObs = 400:10:700;
    sensor = cams(:,37:39);
    colorspace = 'raw';

elseif strcmp(obs,'Aquos5G')
    load('G:\My Drive\Zespri\MATLAB3\phone_sense.mat')
    lambdaObs = 400:10:700;
    sensor = cams(:,40:42);
    colorspace = 'raw';

elseif strcmp(obs,'MiMix2')
    load('G:\My Drive\Zespri\MATLAB3\phone_sense.mat')
    lambdaObs = 400:10:700;
    sensor = cams(:,43:45);
    colorspace = 'raw';

elseif strcmp(obs,'RedMiNote9s')
    load('G:\My Drive\Zespri\MATLAB3\phone_sense.mat')
    lambdaObs = 400:10:700;
    sensor = cams(:,46:48);
    colorspace = 'raw';

elseif strcmp(obs,'SonyExperia1-2')
    load('G:\My Drive\Zespri\MATLAB3\phone_sense.mat')
    lambdaObs = 400:10:700;
    sensor = cams(:,49:51);
    colorspace = 'raw';

elseif strcmp(obs,'SonyExperia5-2')
    load('G:\My Drive\Zespri\MATLAB3\phone_sense.mat')
    lambdaObs = 400:10:700;
    sensor = cams(:,52:55);
```

```
    colorspace = 'raw';

elseif strcmp(obs,'ArrwNX9')
    load('G:\My Drive\Zespri\MATLAB3\phone_sense.mat')
    lambdaObs = 400:10:700;
    sensor = cams(:,55:57);
    colorspace = 'raw';

elseif strcmp(obs,'Pixel4')
    load('G:\My Drive\Zespri\MATLAB3\phone_sense.mat')
    lambdaObs = 400:10:700;
    sensor = cams(:,58:60);
    colorspace = 'raw';
end

if size(sensor,2)~=length(lambdaObs)
    sensor = sensor';
end

%% Formatting (2)

y = lambdaObs;
xq = min(y):1:max(y);

for k=1:3
    x = sensor(k,:)';
    interSen = interp1(y,x,xq);
    newSense(k,:) = interSen;
end
newSense = newSense';

lambdaObs = xq;
sensor = newSense';

% Formatting wavelength ranges ---------------------------------------------
% figure(3)
% subplot(1,2,1),plot(sensor(1,:),'r')
% hold on
% subplot(1,2,1),plot(sensor(2,:),'g')
% subplot(1,2,1),plot(sensor(3,:),'b')
% hold off


lambdaRef = round(lambdaRef(:));
lambdaIll = round(lambdaIll(:));
lambdaObs = round(lambdaObs(:));


lambdaCommon = intersect(lambdaRef,lambdaIll);
lambdaCommon = intersect(lambdaCommon, lambdaObs);

if nbsamples == 1
    ref = ref(ismember(lambdaRef,lambdaCommon));
else
```

```
    ref = ref(:,ismember(lambdaRef,lambdaCommon));
end

ill     = ill(ismember(lambdaIll,lambdaCommon));
sensor  = sensor(:,ismember(lambdaObs,lambdaCommon));
nbBands = length(lambdaCommon);

% %plot sensor
% figure(12)
% subplot(1,2,1), plot(ill,'b')
%
% figure(12)
% subplot(1,2,2), plot(sensor(1,:),'r')
% hold on
% subplot(1,2,2), plot(sensor(2,:),'g')
% subplot(1,2,2), plot(sensor(3,:),'b')
% hold off

if nbBands==0
    disp('Error: no common range.')
    result = [];
    return
end

%% Calculating pseudo-colour

% if strcmp(colorspace,'raw')
%     sensor(1,:) = sensor(1,:)/max(sensor(1,:));
%     sensor(2,:) = sensor(2,:)/max(sensor(2,:));
%     sensor(3,:) = sensor(3,:)/max(sensor(3,:));
% end


%B G R times ill
[~, f1,f2,f3] = deal(lambdaCommon', sensor(1,:).*ill, sensor(2,:).*ill,
sensor(3,:).*ill);

% enforce that basis functions are row vectors
f1 = reshape(f1,[],1);
f2 = reshape(f2,[],1);
f3 = reshape(f3,[],1);

% figure(21)
% plot(f1)
% hold on
% plot(f2)
% plot(f3)
% hold off
%takes (sensor sensitivity * illuminant) gives 100% reflectance
%brightest point in scence - is a white point adjustment

white = (ones(1,size(ref,2))*[f1 f2 f3]);
```

```
% projection
if strcmp(colorspace,'raw')
    rgb = (ref*[f1 f2 f3]);
    rgb = rgb/10;
%     rgb(:,1) = rgb(:,1)/white(1);
%     rgb(:,2) = rgb(:,2)/white(2);
%     rgb(:,3) = rgb(:,3)/white(3);

%     if exist('bd','var')
%         rgb(:,1) = round(rgb(:,1)/white(1)*(2^bd-1));
%         rgb(:,2) = round(rgb(:,2)/white(2)*(2^bd-1));
%         rgb(:,3) = round(rgb(:,3)/white(3)*(2^bd-1));
%     end

%     % RGB -> XYZ
%     barsCIE = barsCIE(ismember(barsCIE(:,1),lambdaCommon),:);
%     barsCIE = barsCIE(:,2:4)';
%
%     M = [sensor(:,1)];
%     xyz = rgb * M;
%     xyz = max(xyz, 0);

else
    xyz = (ref*[f1 f2 f3])/sum(f2);
end

%% Chromatic adaptation

if CAT
    xyz = iCAM06_CAT(xyz*100, white*100, whitepoint('d65')*100)./100;
    xyz = xyz/max(xyz(:,2));
end

%% Output

if strcmp(colorspace,'xyz')

    result = xyz;

elseif strcmp(colorspace,'raw')

    result = rgb;

elseif strcmp(colorspace,'xyY')

    result = [xyz(:,1)./(sum(xyz,2)) xyz(:,2)./(sum(xyz,2)) xyz(:,2)];

elseif strcmp(colorspace,'rgb')

    result = XYZ2SRGB(xyz);
    result = SRGB2RGB(result);

elseif strcmp(colorspace,'srgb') || strcmp(colorspace,'sRGB')
```

```
    %lab = XYZ2LAB(xyz,[1 1 1]);
    result = XYZ2SRGB(xyz);

elseif strcmp(colorspace,'adobeRGB')

    result = XYZ2ADOBERGB(xyz);

elseif strcmp(colorspace,'lab')

    result = XYZ2LAB(xyz,[1 1 1]);

elseif strcmp(colorspace,'linearLAB')

    result = XYZ2linearLAB(xyz,[1 1 1]);

elseif strcmp(colorspace,'lab2000HL')

    result = XYZ2LAB2000HL(xyz,[1 1 1]);

elseif strcmp(colorspace,'lab2000')

    result = XYZ2LAB2000(xyz,[1 1 1]);

elseif strcmp(colorspace,'lch')

    result = XYZ2LCH(xyz,[1 1 1]);

elseif strcmp(colorspace,'lch2000HL')

    tmp    = XYZ2LAB2000HL(xyz,[1 1 1]);
    result = LAB00HL2LCH00HL(tmp);

end

if exist('dims','var')
    result = reshape(result,[dims(1) dims(2) 3]);
end

return

end

%% --------------------------------------------------------------------

function [ill, lambda] = readill(type,startlam,endlam,step)
    switch(type)
        case 1
            tmp = load('ill_D50');
            ill = tmp.data;
        case 2
            tmp = load('Ill_D65');
            ill = tmp.Ill;
        case 3
```

```
            tmp = load('illspec');
            ill = tmp.Illspectra;
%             figure(100)
%             plot(ill)
        case 4
            tmp = load('illD100');
            ill = tmp.Ill;
        case 5
            tmp = load('ill_A');
            ill = tmp.Ill;
        case 6
            tmp = load('ill_F2');
            ill = tmp.data;
            ill = ill(:,[1 2]);
%             figure(100)
%             plot(ill)
        case 7
            tmp = load('whiteCCtile');
            ill = tmp.avgref;
%             ill = ill(:,[1 3])
        case 8
            tmp = load('illFs');
            ill = tmp.Ill;
            ill = ill(:,[1 4]);
        case 9
            tmp = load('illFs');
            ill = tmp.Ill;
            ill = ill(:,[1 5]);
        case 10
            tmp = load('illFs');
            ill = tmp.Ill;
            ill = ill(:,[1 6]);
        case 11
            tmp = load('illFs');
            ill = tmp.Ill;
            ill = ill(:,[1 7]);
        case 12
            tmp = load('illFs');
            ill = tmp.Ill;
            ill = ill(:,[1 8]);
        case 13
            tmp = load('illFs');
            ill = tmp.Ill;
            ill = ill(:,[1 9]);
        case 14
            tmp = load('illFs');
            ill = tmp.Ill;
            ill = ill(:,[1 10]);
        case 15
            tmp = load('illFs');
            ill = tmp.Ill;
            ill = ill(:,[1 11]);
        case 16
            tmp = load('illFs');
            ill = tmp.Ill;
            ill = ill(:,[1 12]);
        case 17
```

```
        tmp = load('illFs');
        ill = tmp.Ill;
        ill = ill(:,[1 13]);
    case 18
        tmp = load('illConcordMK41');
        ill = tmp.Ill;
    case 19
        tmp = load('illCooperDL11C');
        ill = tmp.Ill;
    case 20
        tmp = load('illCooperDL11W');
        ill = tmp.Ill;
    case 21
        tmp = load('illCooperRXD1');
        ill = tmp.Ill;
    case 22
        tmp = load('illCooperRXD2');
        ill = tmp.Ill;
    case 23
        tmp = load('illCREELRP38');
        ill = tmp.Ill;
    case 24
        tmp = load('illCRSMR16');
        ill = tmp.Ill;
    case 25
        tmp = load('illDALBaltic');
        ill = tmp.Ill;
    case 26
        tmp = load('illEnduraWT');
        ill = tmp.Ill;
    case 27
        tmp = load('illEnduraWW');
        ill = tmp.Ill;
    case 28
        tmp = load('illErcoBlue');
        ill = tmp.Ill;
    case 29
        tmp = load('illErcoClear');
        ill = tmp.Ill;
    case 30
        tmp = load('illErcoTest');
        ill = tmp.Ill;
    case 31
        tmp = load('illGCIMR16');
        ill = tmp.Ill;
    case 32
        tmp = load('illGE71');
        ill = tmp.Ill;
    case 33
        tmp = load('illGEF40W');
        ill = tmp.Ill;
    case 34
        tmp = load('illGEFC8T9');
        ill = tmp.Ill;
    case 35
        tmp = load('illGEP30');
        ill = tmp.Ill;
```

```
case 36
    tmp = load('illGEP38F');
    ill = tmp.Ill;
case 37
    tmp = load('illGEP38S');
    ill = tmp.Ill;
case 38
    tmp = load('illGEST');
    ill = tmp.Ill;
case 39
    tmp = load('illGEVIO');
    ill = tmp.Ill;
case 40
    tmp = load('illLaminaTC');
    ill = tmp.Ill;
case 41
    tmp = load('illLaminaWWNB');
    ill = tmp.Ill;
case 42
    tmp = load('illLEDP38');
    ill = tmp.Ill;
case 43
    tmp = load('illLeeliumD');
    ill = tmp.Ill;
case 44
    tmp = load('illLSIC2M2');
    ill = tmp.Ill;
case 45
    tmp = load('illLSIC3M2');
    ill = tmp.Ill;
case 46
    tmp = load('illLSIC4M2');
    ill = tmp.Ill;
case 47
    tmp = load('illLuxinaM250');
    ill = tmp.Ill;
case 48
    tmp = load('illMSIiP');
    ill = tmp.Ill;
case 49
    tmp = load('illNatStud2');
    ill = tmp.Ill;
case 50
    tmp = load('illOsramDia');
    ill = tmp.Ill;
case 51
    tmp = load('illOsramDul');
    ill = tmp.Ill;
case 52
    tmp = load('illOsramP38');
    ill = tmp.Ill;
case 53
    tmp = load('illPhil50P30L');
    ill = tmp.Ill;
case 54
    tmp = load('illPhil50W');
    ill = tmp.Ill;
```

```
case 55
    tmp = load('illPhilBBG480');
    ill = tmp.Ill;
case 56
    tmp = load('illPhilEnd');
    ill = tmp.Ill;
case 57
    tmp = load('illPhilEndB');
    ill = tmp.Ill;
case 58
    tmp = load('illPhilP38');
    ill = tmp.Ill;
case 59
    tmp = load('illPhilTLD');
    ill = tmp.Ill;
case 60
    tmp = load('illPhotonCS5');
    ill = tmp.Ill;
case 61
    tmp = load('illProLiteD');
    ill = tmp.Ill;
case 62
    tmp = load('illSolaisLR38');
    ill = tmp.Ill;
case 63
    tmp = load('illSolux12V');
    ill = tmp.Ill;
case 64
    tmp = load('illSylvaniaC1');
    ill = tmp.Ill;
case 65
    tmp = load('illSylvaniaC2');
    ill = tmp.Ill;
case 66
    tmp = load('illSylvaniaNF25');
    ill = tmp.Ill;
case 67
    tmp = load('illSylvaniaTA');
    ill = tmp.Ill;
case 68
    tmp = load('illSylvaniaUL1');
    ill = tmp.Ill;
case 69
    tmp = load('illSylvaniaUL2');
    ill = tmp.Ill;
case 70
    tmp = load('illSynergy1003605');
    ill = tmp.Ill;
case 71
    tmp = load('illSynergy1003606');
    ill = tmp.Ill;
case 72
    tmp = load('illSynergy1003607');
    ill = tmp.Ill;
case 73
    tmp = load('illSynergy1003608');
    ill = tmp.Ill;
```

```
    case 74
        tmp = load('baso4');
        ill = tmp.baso4(:,[1 2]);
    case 75
        tmp = load('baso4');
        ill = tmp.baso4(:,[1 3]);
    case 76
        tmp = load('baso4');
        ill = tmp.baso4(:,[1 4]);
    case 77
        tmp = load('baso4');
        ill = tmp.baso4(:,[1 5]);
    case 78
        tmp = load('baso4');
        ill = tmp.baso4(:,[1 6]);
    case 79
        tmp = load('baso4');
        ill = tmp.baso4(:,[1 7]);
    case 80
        tmp = load('baso4');
        ill = tmp.baso4(:,[1 8]);
    case 81
        tmp = load('baso4');
        ill = tmp.baso4(:,[1 9]);
    case 82
        tmp = load('baso4');
        ill = tmp.baso4(:,[1 10]);
    case 83
        tmp = load('baso4');
        ill = tmp.baso4(:,[1 11]);
    case 84
        tmp = load('baso4');
        ill = tmp.baso4(:,[1 12]);
    case 85
        tmp = load('baso4');
        ill = tmp.baso4(:,[1 13]);
    case 86
        tmp = load('baso4');
        ill = tmp.baso4(:,[1 14]);
    case 87
        tmp = load('baso4');
        ill = tmp.baso4(:,[1 15]);
    case 88
        tmp = load('baso4');
        ill = tmp.baso4(:,[1 16]);
    otherwise
        ill = rand(1,10);
        windowSize = 5;
        b = (1/windowSize)*ones(1,windowSize);
        ill = filter(b,1,ill);
        if rand>0.5
            ill = ill(end:-1:1);
        end
        ill = [(400:(700-400)/(length(ill)-1):700)' ill'];
end

ill(ill(:,1)>endlam,:) = [];
```

```
    ill(ill(:,1)<startlam,:) = [];
    startlam = ill(1,1);
    endlam = ill(end,1);

    step = max(step,ill(2,1)-ill(1,1));

    lambda = startlam:step:endlam;

    tmp = interp1(ill(:,1), ill(:,2), lambda);
    ill = tmp';

    ill = ill'./max(ill); % normalization
%     figure(2)
%     plot(ill)
%     pause
%
end

% This function takes a set of XYZ values and converts them to AdobeRGB.

function RGB = XYZ2ADOBERGB(XYZ)

    % Load the Adobe RGB profile
    P = iccread('C:\WINDOWS\system32\spool\drivers\color\AdobeRGB1998.icc');

    % XYZ -> RGB, this is the inverse transform, the default transform is:
    % 'Source Color Space -> Connection Color Space' = 'RGB -> XYZ'
    C = makecform('mattrc', P, 'Direction', 'inverse', ...
                  'RenderingIntent', 'AbsoluteColorimetric');
    RGB = applycform(XYZ, C);

end
```