University of Southern Queensland

Faculty of Health, Engineering & Sciences

Utilising a Mobile Device to Aid in Early Diagnosis of Corneal Abnormalities

A dissertation submitted by

Sean Thomson

in fulfilment of the requirements of

ENG4112 Research Project

towards the degree of

Bachelor of your Electrical & Electronic Engineering

Submitted: October, 2019

Abstract

The cornea is the clear front surface of the eye that allows light to enter and provides up to 75% of the eyes focusing power. There are many corneal abnormalities, and early diagnosis for many patients is paramount to preserve sight and avoid corneal transplantation. At this stage, diagnosis predominately relies on specialised topographic imaging equipment and evaluation from an experienced ophthalmologist.

The hypothesis investigated in this research is that an image taken from a mobile device can be reconstructed into a corneal topographic image and used for automatic diagnosis of abnormalities. To prove the hypothesis, this project differentiated between patient images with and without a common corneal condition called Keratoconus (KC).

Through critical analysis of the literature, this research determined machine learning and preprocessing techniques would be adopted. A database of patient images with and without a diagnosis of KC was obtained from the Australian Study of Keratoconus. Preprocessing algorithms and neural networks were developed for this data. The optimum preprocessing techniques and neural network parameters were selected for the final model based on verification with the available database.

The final results were promising as it can differentiate between a patient with and without KC 90.9% of the time. This testing was with unique data not used in the learning process. Additionally, a mobile app was developed, which enabled a user to select an advanced topographic image and observe the models prediction.

This project has shown promising results, indicative that the research hypothesis may be sustainable. Subject to data limitations, further work would show improvements to this research.

This project suggests that additional research with a more extensive database be performed to confirm the results and verify correlations with model predictions and KC severity. Furthermore, if a diverse database of patient images from various topographic imaging equipment is available, more robust classifiers could be created. These classifiers could integrate with mobile topographic technology to further confirm this hypothesis.

University of Southern Queensland

Faculty of Health, Engineering & Sciences

ENG4111/2 *Research Project*

Limitations of Use

The Council of the University of Southern Queensland, its Faculty of Health, Engineering & Sciences, and the staff of the University of Southern Queensland, do not accept any responsibility for the truth, accuracy or completeness of material contained within or associated with this dissertation.

Persons using all or any part of this material do so at their own risk, and not at the risk of the Council of the University of Southern Queensland, its Faculty of Health, Engineering & Sciences or the staff of the University of Southern Queensland.

This dissertation reports an educational exercise and has no purpose or validity beyond this exercise. The sole purpose of the course pair entitled "Research Project" is to contribute to the overall education within the student's chosen degree program. This document, the associated hardware, software, drawings, and other material set out in the associated appendices should not be used for any other purpose: if they are so used, it is entirely at the risk of the user.

Dean

Faculty of Health, Engineering & Sciences

Certification of Dissertation

I certify that the ideas, designs and experimental work, results, analyses and conclusions set out in this dissertation are entirely my own effort, except where otherwise indicated and acknowledged.
I further certify that the work is original and has not been previously submitted for assessment in any other course or institution, except where specifically stated.

Sean Thomson

Acknowledgments

I want to thank John Leis for being my supervisor and offering guidance when required. Notably, prompt feedback towards the end of the research when I was finishing this report. I would also like to thank Dr Srujana Sahebjada and the Centre for Eye Research Australia for their ophthalmologist guidance and data.

I want to acknowledge Karen Zaunscherb for her technical help with developing the mobile app for this research. Without her help there would not have been a functional app in the time available.

I would also like to thank my sister and parents for offering a sympathetic ear and encouragement. However, most of all, I would like to again thank my partner Karen Zaunscherb for her unparalleled and ongoing support. If it were not for her encouragement to pursue this idea I would not have completed my dissertation.

Sean Thomson

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Chapter Overview

This chapter gives the required background knowledge for the project and highlights the aims, objectives, and benefits. This project looks at the automatic diagnosis of corneal abnormalities. Hence, this background knowledge will include a description of the cornea and the targeted corneal condition Keratoconus (KC). Additionally, it presents the fundamentals of machine learning and an artificial neural network (ANN) which are used for this research.

This chapter will expand upon the hypothesis that a mobile device can be used for automatic diagnosis of corneal abnormalities and highlight how this can benefit many patients in the future.

1.2 Background

1.2.1 Cornea

The cornea is the clear front surface of the eye and allows light to enter. Additionally, the cornea provides up to 75% of the focusing power for the eye. The cornea has five layers from front to back these are the corneal epithelium, Bowman's layer, the corneal stroma, Descemet's membrane, and the corneal endothelium (Remington 2011). Figure 1.1 shows

these layers.

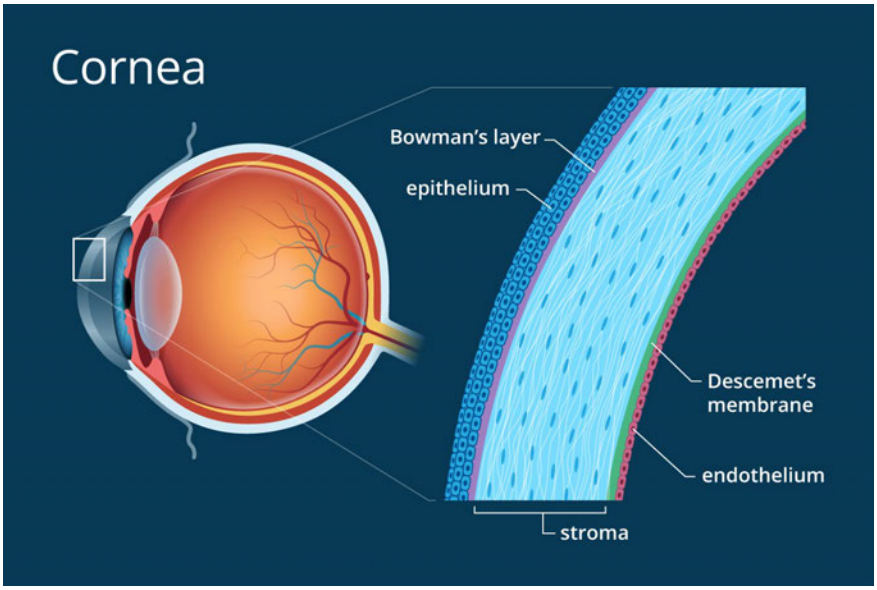


Figure 1.1: Cross-section of cornea (adapted from Heiting (2017)).

The average total centre thickness of the cornea is approximately 550 microns, Figure 1.1 highlights that the stroma makes up most of the this, however, each layer has specialised functions (Remington 2011, Heiting 2017). A brief description of each layer is as follows:

The corneal epithelium. This layer is about 50 microns thick and makes up slightly less than 10% of the total cornea. It enables the eye to keep healthy and moist to enable clear, stable vision by being an optimal surface for the tear film to spread over the eye.

Bowman's layer. This is a skinny layer, only getting up to 14 microns thick. It is a fibrous sheet of connective tissue which forms the transition between the corneal epithelium and stroma. This layer helps prevent corneal scratches from penetrating the stroma. A scratch which is limited to the outer epithelial layer can heal without scarring, whereas, a scratch of the stroma can leave permanent scars and affect vision.

The corneal stroma. This layer is approximately 500 microns and is 90% of the corneas total thickness. It is strands of connective tissue called collagen fibrils, and these are arranged parallel to the corneal surface in bundles across the entire cornea. The uniform spacing and regular arrangement of this layer enable the cornea to be clear.

Descemet's membrane. This is another skinny layer. The thickness of this layer varies throughout life, from 5 microns in children to approximately 15 microns in adults.

Corneal endothelium. This single layer of cells is only 5 microns thick and fills

the space between the cornea and the iris and pupil. This layer maintains the fluid contained within the cornea. Damage to the corneal endothelium can cause swelling affecting corneal health and vision.

Many conditions affect the cornea. Some common ones include dry eyes, corneal ulcers, corneal abrasion, arcus senilis, and corneal dystrophies. Routine eye examinations can result in the preliminary diagnosis of corneal conditions. Diagnosis is then confirmed through in-depth clinical evaluation and a variety of tests.

These corneal tests by an ophthalmologist can include refraction exams, a keratometer, advanced topography systems, or a slit lamp examination where the eye is viewed under a microscope (Wadsworth 2019). This research project targets a condition called KC.

1.2.2 Keratoconus

KC is classically a noninflammatory condition characterised by gradual changes of the cornea's shape. Patients with KC experience progressive thinning of the cornea, which may eventually cause the cornea to protrude forward in a conical shape or bulge. This forward protrusion can lead to blurry vision, photophobia, and other vision problems. Often this begins at puberty; however, the specific causes of the condition remains unknown. It is believed that genetic factors do play some role (Wadsworth 2019) and it is a bilateral and asymmetric condition usually characterised by a slow progressive evolution (Galvis, Sherwin, Tello, Merayo, Barrera & Acera 2015).

In the early stages of the disease, KC is managed predominantly with glasses and contact lenses. The past decade has seen the introduction of collagen cross-linking (CXL), which slows KC progression by stiffening the cornea. This surgical procedure requires maximal corneal thickness to protect the corneal layers and is only suitable for patients in early stages of KC (Kymes, Walline, Zadnik & Gordon 2004). Early detection is, therefore, a prerequisite for this treatment.

Once KC progresses, patients inevitably require corneal transplantation. KC patients comprise the second largest group of transplant patients, requiring 27% of all corneal transplants. However, transplantation is not a panacea, while corneal transplants are able to restore vision, in 20% of cases the grafts are rejected, and patients require re-grafting

and a lifetime of follow-up care (Gain, Jullienne, He, Aldossary, Acquart, Cognasse & Thuret 2016).

The high demand for corneas has resulted in a worldwide shortage of corneal donor tissue, and approximately 12.7 million people are on the corneal transplantation waiting list (Gain et al. 2016). This has resulted in the disease coming to the attention of research centres as solutions are needed for early diagnosis (Lavric & Valentin 2019).

Early diagnosis of KC currently relies on the use of specialised topographic cameras and then careful analysis by an experienced ophthalmologist. As this equipment is not easily portable, early diagnosis may not be readily available especially in the developing world.

1.2.3 Machine Learning

Machine learning is a subset of artificial intelligence (AI) that is applied to data. Machine learning provides a system with the ability to learn and improve from experiences without being explicitly programmed to do so. Healthcare began to use machine learning earlier than most fields due to humanitarian needs and medical research progress. Thanks to the growth in the last decade machine learning is expanding as an immediate opportunity in health care, including eye care and vision (Catania & Nicolitz 2018, Shobha & Rangaswamy 2018).

An ANN is a set of algorithms used in machine learning for modelling data using graphs of neurons. ANNs are loosely modelled after the neuronal structure of the human brain.

The network is organised in layers, and they consist of several interconnected nodes or neurons. Patterns are presented to the ANN at the input layer, which communicates with the hidden layers and processing occurs via weighted connections (Tang, Fong, Deb, Vasilakos & Millham 2018). The hidden layers link to the output layer, and the network provides an answer. Figure 1.2 shows a traditional ANN architecture.

In supervised learning, the ANN is presented with labelled data. It calculates a random prediction and then based how wrong it is from the correct answer it makes appropriate adjustments to its weights. During the learning process, the network requires lots of data, and it will iterate many times until the guess is as close as possible to the exact answer (Tang et al. 2018).

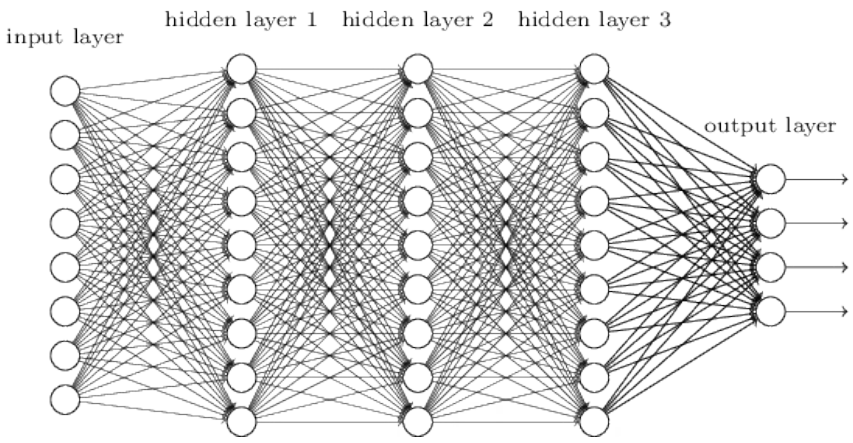


Figure 1.2: Traditional ANN model. Every neuron in one layer connects to all neurons of the next (adapted from MissingLink (2019)).

After the learning process, the ANN can predict on unique data. ANNs are extensively used throughout this project and will be further explained in Chapter 2 and 3.

1.3 Aim and Objectives

The project aim is to determine if an ANN and a mobile device can be used to help diagnose corneal abnormalities.

The hypothesis is that an image taken from a mobile device can be reconstructed into a corneal topography image and can be utilised for automatic diagnosis of abnormalities.

Diagnosis would be without the need for an extensive review from the ophthalmologist or specialised imaging equipment. As corneal abnormalities are quite prevalent in the developing world and frequent access to an ophthalmologist is rare, it would be advantageous if any medical professional could give a preliminary diagnosis and begin treatment or get specialised care for their patient. Additionally, if the specialised equipment is available, the automatic analysis by an ANN would enable a preliminary diagnosis to occur quickly. Software analysis could alleviate the errors in the interpretation of patient conditions as diagnosis with specialised equipment can still be subjective and requires an experienced ophthalmologist.

The conditions and abnormalities that can affect the cornea are extensive. As such, this project looked at common corneal dystrophies. The most common are KC, Fuchs's

dystrophy, lattice dystrophy, and map-dot-fingerprint dystrophy. This project is further refined and focuses on diagnosing KC. The research will determine if automatic diagnosis of KC is feasible using an image from a Pentacam® and investigate adapting this for a mobile device.

The results from automatically detecting KC in this project will indicate the potential of implementing similar techniques to diagnose different corneal abnormalities. Additionally, it will show if further research should be committed to implementing similar detection software for a range of corneal imaging equipment. As success in automatically diagnosing KC from a Pentacam and other imaging equipment is the first step in a standalone mobile solution, this research will also investigate the feasibility of creating a corneal topographic image from a mobile device.

For this system to be feasible, it needs to show it can accurately identify KC from a corneal topographic image. For this aim, the following broad objectives must be met:

(i) Obtain patient images of corneas with and without KC.

(ii) Process images in preparation for ANN learning.

(iii) Train ANN on processed patient images.

(iv) Test performance of ANN on data not used for learning.

(v) Investigate integrating results from the project for a mobile device.

The objectives will be further expanded in the methodology section shown in Chapter 3.

1.4 Outcomes and Benefits

This research project will help determine if utilising ANNs are suited to the detection of corneal conditions. It will highlight an optimum solution within the constraints of this project and discuss if a smart mobile can replace the use of specialised equipment. Furthermore, it will help show if corneal abnormalities can be autonomously detected without the need for extensive review from the ophthalmologist. If it is successful, it can help medical professionals diagnose serious and challenging diseases.

The expected outcomes of this project will be to:

- increase the understanding of ANNs for KC detection.
- improve confidence in utilising computer vision for corneal abnormality detection.
- show the versatility of a smart mobile.
- encourage further research in this area.

If successful, these outcomes would show direct benefits for patients with KC. Sahebjada et al. (2014) showed that the quality of life of KC patients is subsequently lower than that of patients with later onset eye diseases such as age-related macular degeneration or diabetic retinopathy. This highlights the significant long-term morbidity associated with KC. If an early diagnosis of a patient is achieved, treatments like CXL would be available, which would significantly increase patient quality of life.

Additionally, KC represents 31% for corneal transplantation in Australia, and patients typically require up to five transplants in their life (Williams, Lowe, Keane, Jones, Loh & Coster 2015). Again, early diagnosis is pivotal in reducing the number of patient transplants and by virtue, reducing extensive corneal transplant waiting lists.

Ultimately, if success can be seen for KC patients, this research can be adapted to many other corneal abnormalities in the future. It is reasonable to assume that most medical professionals will have access to a smart mobile and hence, the ability to use an extension of this project. This can benefit any patient without access to an ophthalmologist with specialised equipment.

1.5 Chapter Summary

This chapter has explained that this project will aim to show a diagnosis of KC is achievable using topographic images and that a mobile device can process the image. Also, this project seeks to encourage further research into KC detection and adapting these techniques for other corneal abnormalities.

Chapter 2

Literature Review

2.1 Chapter Overview

There are some critical areas of research to determine the suitability and knowledge gap for this project. The areas of technical analysis include machine learning, ANNs, techniques for training networks, digital signal processing (DSP) on smart mobiles, and retina scanners. Additional research will consist of diagnosing KC, corneal imaging techniques, and the outputs of a Pentacam.

The technical research areas are extensive. As such, the literature review will specifically focus on cornea abnormalities, eyes and medical conditions.

While there are other DSP techniques apart from machine learning used to identify areas of interests within images, these will not be investigated. As will be further shown in this literature review machine learning is the right solution for this project.

2.2 Corneal Imaging

This section will explain the fundamentals of the conventional imaging techniques corneal topography and tomography. It explains common topographic patterns that can indicate a patient with KC and how corneal tomography is helping with early diagnosis.

2.2.1 Topography Fundamentals

As there is mention of corneal topography throughout this report, it is imperative the fundamentals be understood. As mentioned, the cornea is important to vision since changes in the cornea's shape also affect visual acuity. There are many ways to estimate the corneal surface, like, keratoscope, interferometry, and triangularisation (Klein 1997). As current literature, shown by Section 2.11, will highlight that the keratoscope is the most common solution for a portable attachment used in conjunction with a mobile device, it was further researched.

The corneal surface reconstruction utilising a keratoscope is also commonly known as Placido's disc and can be summarised in a few steps: image capture from reflected concentric rings on subjects' eye, feature extraction via image processing, and surface estimation using reconstruction algorithms (Klein 1997).

The first step is to capture the corneal image, and the reconstruction of the surface is based on how the cornea reflects light. The pattern captured in the image is from normal reflections. This is important as it means the reflection type will follow the law of reflection. The law of reflection states that the angle of reflection to the surface normal equals the angle of incidence to the surface normal. See Figure 2.1 and Equation 2.1.

$$\theta_r = \theta_i \tag{2.1}$$

From this law, corneal surfaces with contrasting norms will reflect the incident ways differently. When capturing the image, the location of the emitter and camera are known. Hence, this information can be used for surface reconstruction (Rosa 2013).

In this keratoscope approach, the camera is placed behind and in the middle of the Placido disc and aligned with the centre of the eye. The reflected image from the cornea can then be received. With this configuration, eyes with normal corneal curvature produce circles in the captured image and KC will produce deformed rings while other astigmatisms caused by different corneal curvatures produce ellipses. Figure 2.2a shows the Placido disc and simple examples of captured images before feature extraction and surface estimation.

Concluding the image capturing step, the image processing step and surface estimation step follow. Image processing step involves associating rings within the emitter to the captured image to identify the pattern. To improve identification image enhancing techniques

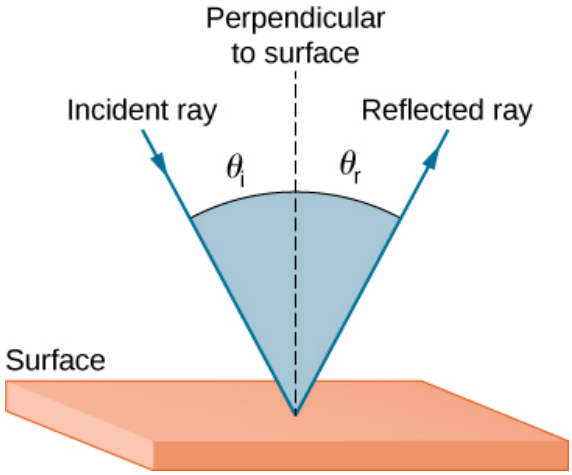


Figure 2.1: The law of reflection states that the angle of incidence equals the angle of reflection $\theta_i = \theta_r$. Measured relative to the perpendicular to the normal surface where the ray strikes (adapted from Ling et al. (2019)).

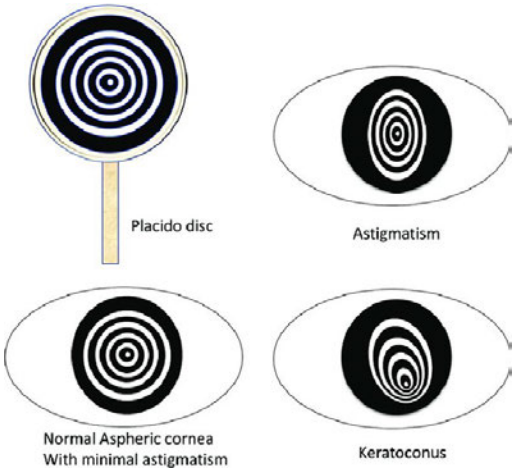
are used to remove noise, equalise brightness, and close holes in the design. Concluding image extraction, there is the surface estimation step. The surface estimation will discover a surface that approximately generates a reflection consistent with the data, following the laws of reflection (Rosa 2013). Figure 2.2b shows an example of a reconstructed corneal surface utilising the keratoscope approach. Many techniques can be used for image processing and surface estimation. These will be explored further in Section 2.2.2, 2.2.3, and 2.4.

Computer-generated corneal topography images aid ophthalmologist's in diagnosing corneal dystrophies and other related eye disorders.

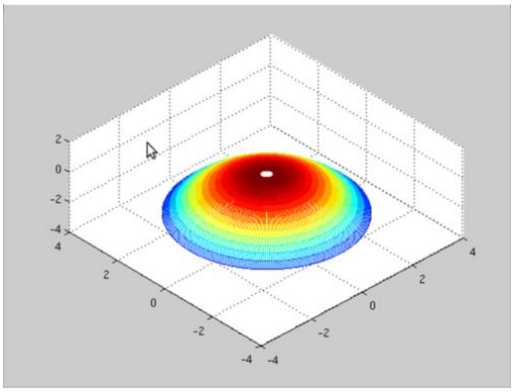
2.2.2 Corneal Topography

Corneal topography is a non-invasive imaging technique used for mapping the outer surface curvature of the cornea. It is an advancement from its predecessor the keratoscope, which, enabled four measurements over a few millimetres. Corneal topography is now a grid of thousands of points covering the entire cornea (Schirmer 2003).

Corneal topography is obtained by focusing light on the anterior surface of the patient's cornea and capturing the reflected image with a digital camera. The reflection pattern back to the camera represents the topology of the cornea. A computer is then used to provide analysis and determine the heights of thousands of points across the cornea. This



(a) Placido disc and representative patterns of corneal shapes (adapted from Fan et al. (2018)).



(b) Example of reconstructed corneal surface with minimal astigmatism (adapted from Rosa (2013)).

Figure 2.2: Corneal topography utilising Placido disc.

topographical map can be represented in several ways. However, a common technique is to colour-code the steepness of curvature (Martínez-Abad & Piñero 2017).

Patients with topography maps with an asymmetrical bowtie pattern or astigmatism (blurred vision) are candidates for KC. Corneal abnormalities or deviations from symmetrical bowtie patterns are detected by the curvature map and are described by their shape. Examples of these descriptions include oval, irregular, superior/inferior steep, symmetric bowtie with a skewed radial axis, and inferiorly steep asymmetric bowtie. Patients with these patterns are risk factors for KC when accompanied by abnormal corneal tomography (Tur, MacGregor, Jayaswal, O'Brart & Maycock 2017).

While corneal topography relies on reflection of light from the anterior cornea, corneal tomography also provides a measurement of the posterior shape of the cornea.

2.2.3 Corneal Tomography

Corneal tomography has allowed earlier detection of KC and other corneal ectasias as it presents a quantitative examination of both the anterior and posterior corneal surfaces. Examining the posterior corneal surface can highlight new elevation changes, stromal thinning, and ectasia (Tur et al. 2017).

2.3 Diagnosing Keratoconus

KC is a bilateral and commonly asymmetrical disease in which the cornea becomes conical in shape. Typically, it presents in adolescence and progresses until approximately the fourth decade of life. KC is one of the most common causes of keratoplasty in the developing world (Martínez-Abad & Piñero 2017). KC has a significant association with eye rubbing, contact lenses, allergic eye diseases, and repeated trauma. Furthermore, a review of KC diagnosis highlighted that 8% to 10% of cases have a hereditary component and family aspect (Tur et al. 2017).

As KC progresses, there is tissue loss and a loss of correlation between posterior and anterior corneal curvature. Figure 2.3 shows the anterior and posterior stroma concerning the rest of the cornea and eye.

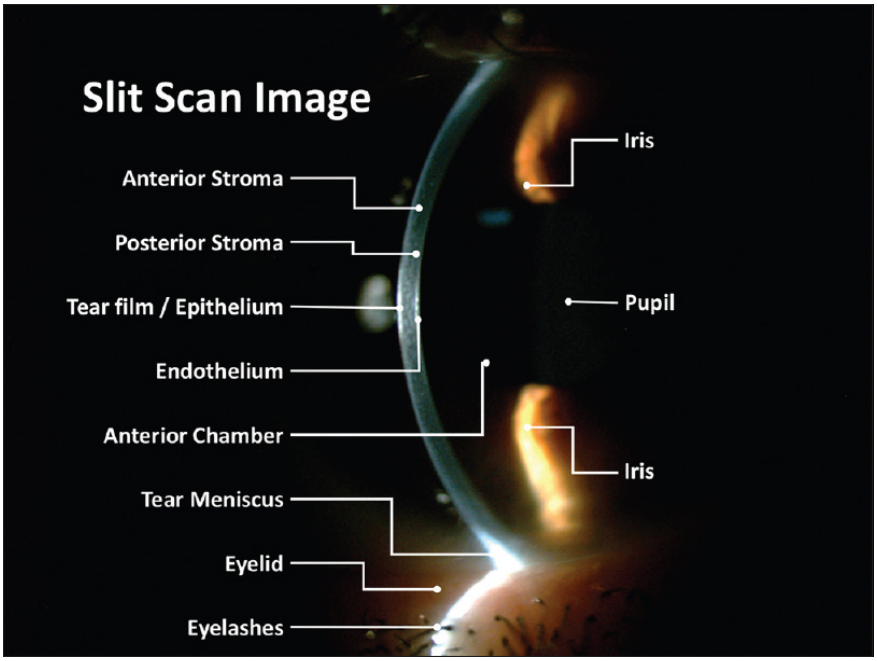


Figure 2.3: Cornea and anterior eye assessment slit scan image (adapted from Martin (2018)).

KC diagnosis is commonly achieved via slit-lamp examination, measurement of visual acuity, or corneal topography. Since corneal thinning and distortion causes a cone or conical-shaped protrusion in KC patients, this may be visible utilising a slit lamp examination in advanced cases. However, as visual acuity may not be affected and when KC is in its infancy, it is not detectable by slit lamp exams the disease may go undiagnosed unless an assessment of the posterior and anterior corneal surfaces is undertaken using corneal tomography or topography (Tur et al. 2017).

Recent advances in corneal imaging, giving the potential to assess the corneal surface with the aid of anterior and posterior elevation measurements have provided ophthalmologists with valuable information. The Pentacam utilises the Scheimpflug imaging technique to present corneal topographic indices with excellent repeatability and accuracy. KC diagnostic indices and pachymetry (thickness) indices are among the index measured by the Pentacam, which facilitate the diagnosis (Hashemi, Beiranvand, Yekta, Maleki, Yazdani & Khabazkhoob 2016). The Pentacam and typical results will be discussed further in Section 2.4.

2.4 Pentacam

Imaging techniques like corneal topography are crucial for diagnosing and treating a wide variety of ocular diseases, including KC. The OCULUS Pentacam utilises Scheimpflug technology to create topographic maps and reports (Hashemi et al. 2016). A typical output of the Pentacam is the four maps report. Figure 2.4 shows an example.

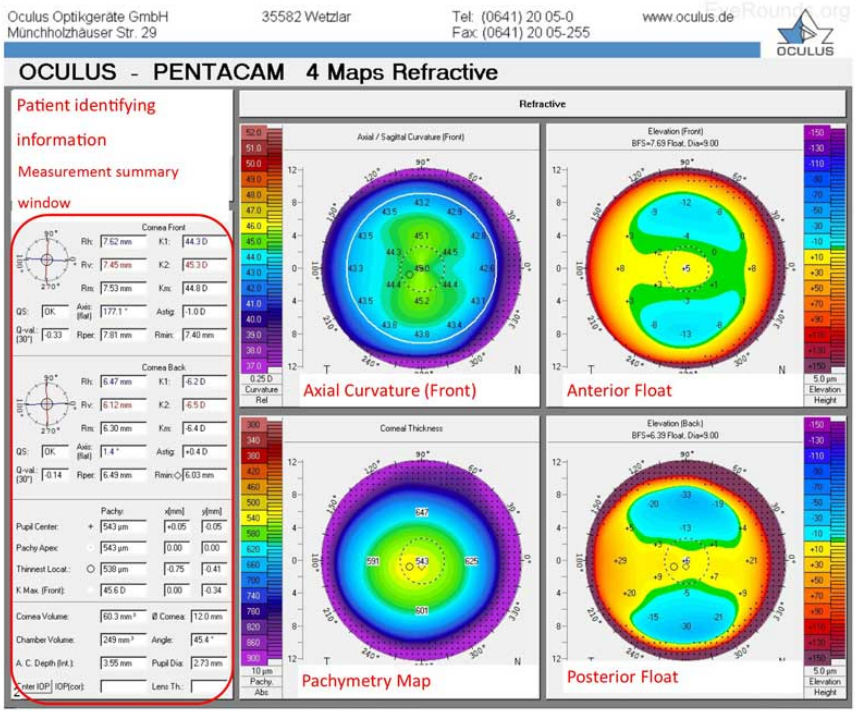


Figure 2.4: Pentacam four map report (adapted from Greenwald et al. (2016)).

Figure 2.4 illustrates the axial curvature map, also known as the sagittal map, shows the curvature of the anterior corneal surface in dioptric values. Cooler colours represent flatter areas, and warmer shades are steeper. For the anterior and posterior float (eleva-

tion), cooler colours denote where the cornea is depressed below the best fit sphere, and warmer tones indicate where the cornea is elevated above the best fit sphere. Lastly, the pachymetry map is a colour map which shows corneal thickness; cooler tones are thicker and warmer colours thinner (Greenwald et al. 2016).

The Pentacam can also provide eight map reports, three-dimensional representation of the cornea, and many raw data points. However, as the available data for this project will be of the four maps report, and this is adequate for diagnosing KC, the other Pentacam features are not discussed further. Figure 2.4 is an example of an image that is used to isolate each coloured map for further processing.

2.5 Machine Learning

Artificial Intelligence (AI) was introduced in the fifties. Over the following decades, the general interest grew, along with its applied use to data, referred to as machine learning. Healthcare began to use AI earlier than most fields due to humanitarian needs and medical research progress. Thanks to the growth in the last decade machine learning is expanding as an immediate opportunity in health care, including eye care and vision (Catania & Nicolitz 2018, Shobha & Rangaswamy 2018).

A machine-learning algorithm was developed to detect corneal abnormalities using the principle of pattern detection through corneal topography. Additionally, machine learning has been used to extract details from infrared images of the ocular surface to diagnose dry eye disorders. While machine learning and ANNs are in their early stages, they are already outperforming the passive, traditional computer vision systems (Shobha & Rangaswamy 2018).

Evidence of AI's success is when ANNs were applied to diagnosis and provide treatment advice for cataracts. Results of 90% were achieved, and this exceeds that of the hospital's ophthalmologists (Catania & Nicolitz 2018).

Machine learning classifiers (MLC) have been employed to detect subtle defects in eyes from patients with human immunodeficiency virus (HIV). The hypothesis was that patients with HIV have retinal damage that causes subtle defects when compared to healthy subjects. Supervised MLC techniques were validated, as it uncovered differences not dis-

coverable by human experts (Kozak, Sample, Hao, Freeman, Weinreb, Lee & Goldbaum 2007).

Machine learning has been successfully used to supplement, and sometimes exceed professional medical diagnosis. There are indications that it can be used to detect corneal abnormalities, although this is yet to be successfully implemented. To further understand computer vision using machine learning, a fundamental understanding of ANNs is imperative.

2.6 Artificial Neural Network

There are many interpretations and variance on ANNs. The standard and widely accepted ANN learning paradigms are explained here. There are three machine learning algorithms associated with ANNs, supervised, unsupervised, and reinforcement (Li, Chen, Wang, Zhou & Tang 2019, Lin, Wang & Hao 2017).

In supervised learning algorithms, labelled data is presented to the machine. The ANN is shown many paired inputs with desired outputs. The learning task is to produce the desired outputs that, on average, reduces the error for all data. The mean squared error is commonly used as the function to determine the cost between input and desired output. The algorithm will iteratively make predictions and correct itself based on the cost. This learning will progress until an acceptable level of performance (Tang et al. 2018, Lin et al. 2017).

With Unsupervised learning algorithms, the machine interacts with unlabelled databases. In this paradigm, the ANN has input data with no corresponding output variable. This learning aims to find a model that represents the underlying structure or distribution in the data. Unsupervised learning can be more problematic than supervised learning. However, as the machine is left to their own devices, it can discover impressive structures to the data (Sanger 1989).

Reinforcement learning is where the machine utilises a technique called memory networking. The aim is to weight the network, so it performs actions that minimise the longterm cost. When the device conducts an act, the environment generates a result and an instantaneous cost. This cost is according to some rules. The machine will explore

the compromise between new actions uncovering their cost and exploit prior learning to progress further and quicker (Li et al. 2019).

There are ANN models discussed that use unlabelled data for supervised learning and labelled data for unsupervised (Catania & Nicolitz 2018). However, the ANN interpretation presented here is usually accepted as the norm.

While this "black box" approach of supplying an algorithm large data sets and allowing it correlate results does dismay some. It is hard to argue against performance. This has changed the approach to some computer vision problems like object detection and recognition. While there can be limitations to this approach, if for example, large sets of training data are not available or the time to train an algorithm is unacceptable. However, it is outperforming some traditional approaches like statistical pattern recognition (Chellappa 2016).

A limitation for ANNs is during the training of the classifier. Should the database for training not be sufficiently broad or if the system rapidly converges to a solution, the results may not be acceptable. As accuracy within the medical fields is paramount and an extensive database is not always available, there need to be some techniques explored for training.

2.7 Techniques for Training Artificial Neural Networks

ANNs are efficient at solving numerous problems. However, finding practical training algorithms or large datasets can be challenging. Large datasets are pivotal as typically the available data would be segregated into three groups a training dataset, a validation dataset, and a test dataset.

The training dataset is used to fit the parameters, for example, the weights and biases. Unfortunately, a training dataset used alone will tend to identify apparent relationships in the data that do not hold in general, called overfitting. A validation dataset is used to compare the performance of the learning algorithms. It can be used in a process called early stopping if overfitting of the training data is observed. Finally, an independent dataset is used to test the network. If a model fits the test and training dataset, then minimal overfitting has occurred (Bishop et al. 1995). In the absence of large datasets,

there are alternative methods for training classifiers.

There is a recent algorithm called dynamic group optimisation (DGO). This algorithm has a strong performance in searching for and avoiding local optima with rapid convergence. Furthermore, a hybrid solution which combines feedforward neural networks (FNN) and DGO (Tang et al. 2018) can present reliable results.

DGO can be used to improve the problem-solving abilities of FNN. Two solutions for a hybrid model include using the DGO to find an optimum structure of an FNN or using the DGO to tune weights and biases. The results have outperformed other algorithms, like backpropagation, in three examples. The proposed solution is simple and easy to hybridise into other applications (Tang et al. 2018).

One problem with conventional training is that the cost will decay over time. In these traditional training methods, users are required to define initial parameters and training standards, if this is not accurate, it can result in inadequate local solutions. An adaptive learning rate can aid in alleviating this problem (Takase, Oyama & Kurihara 2018).

The learning rate will increase or decrease adaptively so that the training loss can be reduced. Figure 2.5 shows a simple representation of this adaptive learning which adaptively increased between 50 and 100 training epochs.

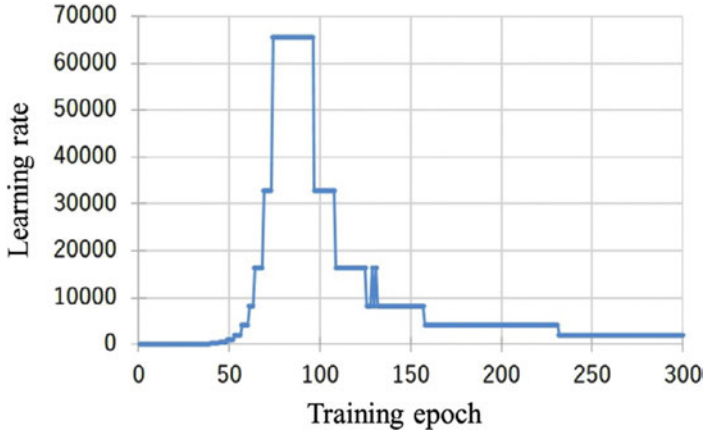


Figure 2.5: Transition in learning rate during one trial (adapted from Takase et al. (2018)).

This approach does show promising results, the most significant being that the solution is more robust, as it is less dependent on initial conditions. However, it did only outperform other methods in specific requirements (Takase et al. 2018). While this did not yield amazing results, it is worth considering as it would be an improvement for some datasets.

ANNs have shown promising results in medical image analysis; however, due to a lack of large datasets, the accuracy and reliability are restricted. Although using segmentation networks can present a solution (Wong, Syeda-Mahmood & Moradi 2018).

Medical image classifiers have been established with limited data using a segmentation network that is pre-trained on similar data classification. It first learned a more comfortable shape and structural concepts before tackling the more complicated problem at hand. This showed that utilising this method for brain tumour classification could give high performance with limited data (Wong et al. 2018). The results outperformed a framework using ImageNet with pre-trained classifiers. This model highlights that there are methods available in the absence of large data sets.

This section shows some new techniques that can be used to train ANNs. It highlights that in the absence of perfect data there may still be potential solutions. Following the training of the classifier, it is necessary to investigate its integration onto a smart device, shown by Section 2.9 and 2.11.

2.8 Keratoconus Diagnosis with Corneal Topography

KC detection algorithms using convolutional neural networks has been proven possible (CNN) (Lavric & Valentin 2019). The corneal topography, which is often interpreted by an experienced ophthalmologist is the input for a detection algorithm during the learning process associated with the CNN. The algorithm involves image preprocessing step which decomposed each image at the pixel level, creating a three-dimensional matrix. Figure 2.6 shows the structure of the CNN used by this research paper.

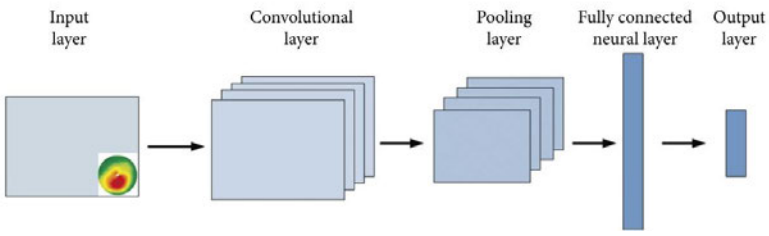


Figure 2.6: Structure of convolutional neural network (adapted from Lavric & Valentin (2019)).

Figure 2.6 shows that the image passes through a series of kernel convolutional filters,

pooling layers, and neural layer. The output gives a normalised classification from 0 to 1, with 1 being a high correlation with KC and 0 no correlation. The image shown within the input layer is an example of a patient's corneal topography with KC. The colours used show any changes to the elevation parameter. Hence, reds indicate the existence of high elevation whereas, light colours like greens and yellows indicate a uniform distribution of height. While using CNN for KC has shown adequate results, it was done with a lack of data. Furthermore, the corneal topographies were modelled and not patient images (Lavric & Valentin 2019).

2.9 Digital Signal Processing with Smart Devices

Smart mobile devices such as telephones and tablets are prevalent due to their powerful hardware and features. In 2019 the prediction is that there will be 2.5 billion smartphone users not including tablets (Liu, Ning, Mu, Zheng, Zeng, Yang, Huang & Ma 2017). A medical application can be adapted for these devices.

Smart mobile devices are equipped with high-speed processors, high-resolution cameras, and numerous peripherals; hence, their use is becoming widespread. The target audience for medical applications utilising smartphones could extend to around a third of the world.

An inexpensive wireless blood pressure sensor has been used with a mobile device to determine blood pressure. This signal processing technique within a smartphone yielded results of around 94% accuracy when compared to actual results from a sphygmomanometer (İlhan, Yıldız & Kayrak 2016).

A voice analyser (VA) program was developed for windows to assess voice quality and hence help diagnose voice disorders. Voice clinicians used this program as another tool to carry out perceptual voice evaluations. The aim was to improve the windows VA and to develop a VA smartphone program for broader use. A strong correlation between the smartphone and computer versions, with respects to the signal-to-noise ratio and fundamental frequency were found (Kojima, Fujimura, Hori, Okanoue, Shoji & Inoue 2018).

A smartphone-based immunosensor with nanoflower signal amplification was used for the detection of Salmonella in dairy foods and water. It was observed that the smartphone

device gave similar accuracy and sensitivity compared to the standard microplate reader. A smartphone to detect foodborne pathogens is a convenient alternative to the expensive microplate reader (Zeinhom, Wang, Sheng, Du, Li, Zhu & Lin 2018).

This section shows the versatility and power of modern-day smart devices. The processors of a modern mobile device can handle some complex digital signal processing tasks.

2.10 Retinal Scanners

Detecting abnormalities on corneas using computer vision is still in its infancy, whereas, retinal scanners have been widely employed. Some methods utilised could perhaps be adapted for this project.

An unsupervised retinal vessel segmentation system was developed to help diagnose patients with the fundus disease. It works on the green channel of a red, green, and blue (RGB) image as this has the highest contrast between vessels and background. It uses the direction map of a retina scan image assigning each pixel one out of twelve discrete directions. The image is segmented into seven by seven-pixel models, and twelve plans cover all possible vessel directions. Figure 2.7 shows the discrete direction and final result.

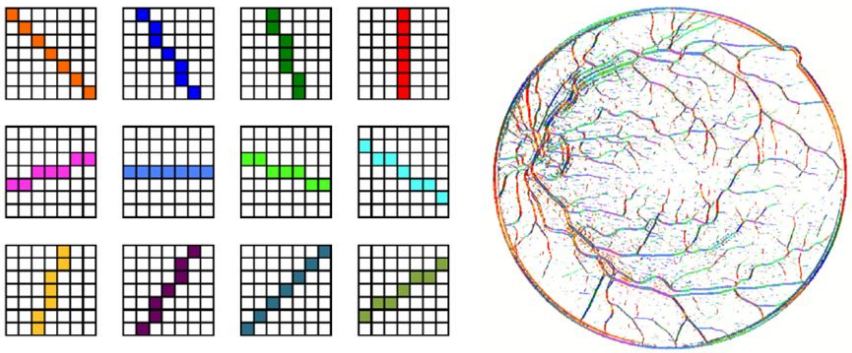


Figure 2.7: Twelve discrete direction (left) and retina vessel map with colour denoting direction (right) (adapted from Frucci et al. (2016)).

Figure 2.7 shows a different colour was used for each discrete direction to give a holistic direction map of the retina. The image on the right of Figure 2.7 is quite noisy. The next process is to utilise filters to remove components with small sizes and peripheral regions (Frucci et al. 2016).

This retinal scanner presented satisfactory results in terms of precision, sensitivity, and accuracy when compared with other conventional preprocessing methods. However, where it excelled was with a computational advantage, as it does not require any preprocessing.

The design of a recognition system for retinal images using ANNs was implemented. The stages of recognition of the retina are image acquisition, feature extraction, and classification of each feature. The ANN was trained using backpropagation, and it was implemented using MATLAB®. The retina recognition system used was based on a three-layer ANN (Sadikoglu & Uzelaltinbulat 2016). Figure 2.8 shows a representation of the model.

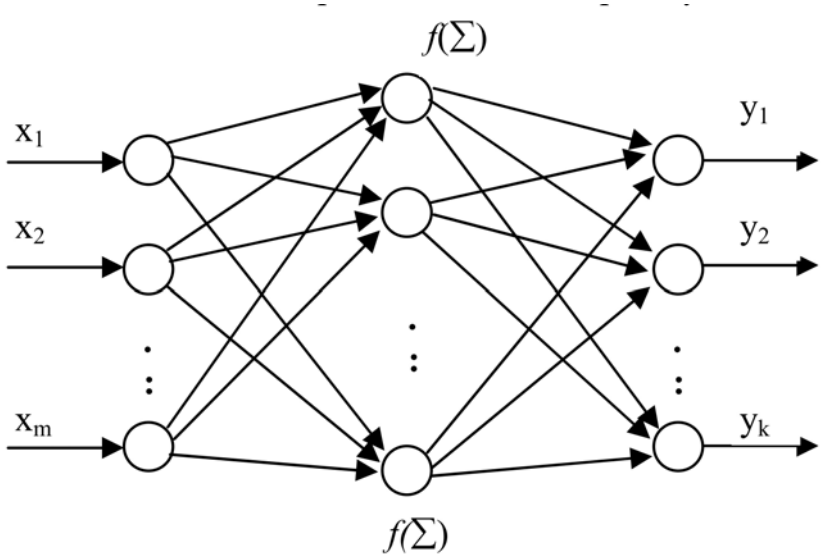


Figure 2.8: Multilayer feed-forward network (adapted from Sadikoglu & Uzelaltinbulat (2016)).

Figure 2.8 shows the input (x), hidden, and output (y) of the three-layer ANN. This image is a simplified version of how an ANN is connected. In a standard ANN, there would be significantly more neurons shown. The system was trained using a dataset of retina images available online. The number of hidden neurons was varied from 8 to 35, with an accuracy improving from 85% to 97.5% (Sadikoglu & Uzelaltinbulat 2016). This did yield impressive results compared to other literature. However, controlled data was used in contrast to actual patient data, and the ANN was relatively simple.

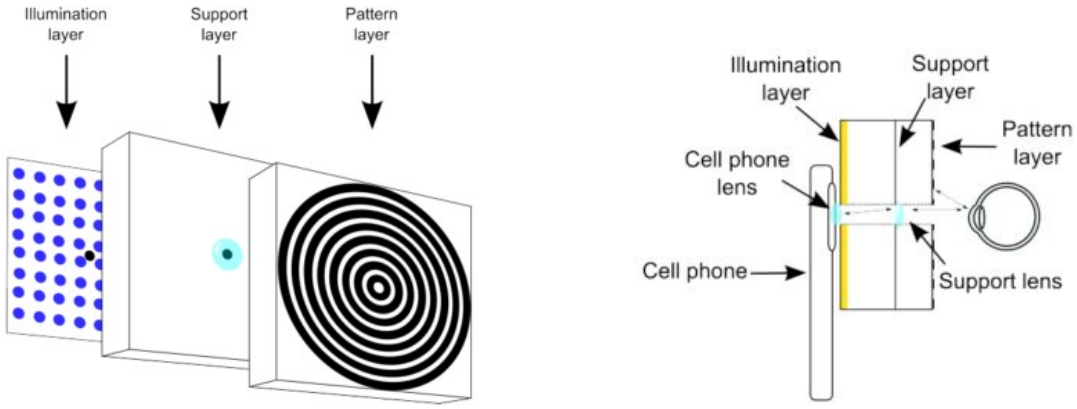
This section highlights some different techniques that may be worth adapting throughout this project.

2.11 Corneal Topography Mapping with Mobile Devices

There is a patent for an apparatus attached to a mobile device which enables corneal topography images utilising a Placido disc illumination system. The Placido disc illumination system is attached to the device's camera and creates concentric rings reflected off the cornea. The camera captures the reflected images which are used to confirm vertex distances (Wallace & Sarver 2007).

The mobile device needs to have memory, a processor, a camera, and the software. The patent states that the captured image if the quality is above a threshold is uploaded to a cloud-based server to generate a corneal topography map (Wallace & Sarver 2007). This patent does show potential. Ideally, to further expand on this project, a corneal topography image created on a mobile device is beneficial. Future trained models on a mobile device can then process the topographic image.

A prototype for a clip-on device to be used in conjunction with a mobile phone to capture reflections from a Placido disc was created and implemented (Rosa 2013). The method created consisted of three layers: the illumination layer, support layer, and the pattern layer. Figure 2.9 shows the conceptual prototype.



(a) Layers. (b) Side view.

Figure 2.9: Conceptual corneal topography model with mobile phone (adapted from Rosa (2013)).

Referring to Figure 2.9. The illumination layer was a series of light-emitting diodes to provide illumination. The support layer helped with image acquisition by using a lens and reduced light dispersion. The last layer was the pattern layer giving the shape to the pattern. While this was an inexpensive model that yielded adequate results again, the

image processing was on a computer. However, as this model was designed in technology respects a long time ago, it is not unreasonable to assume that a modern mobile phone with improved camera and processing power could handle this problem and yield better results.

This section shows that there is potential to utilise a mobile phone to construct a corneal topography estimation. It may also be beneficial to use a cloud-based application in the future for intensive processing, especially as the world becomes increasingly connected. Although, there still lies the challenge, whether this would be accurate enough to use in conjunction with an ANN.

2.12 Chapter Summary

This literature review shows there has been extensive work with ANNs and medical diagnosis. The results obtained from these studies are promising. However, there is limited research in corneal abnormality detection, and the work that has been achieved is with limited data and in some cases, questionable results.

This chapter has also highlighted numerous techniques that can be used for this project and has given a fundamental understanding of diagnosing KC and the topographic imaging that is required. Furthermore, topographic imaging with a mobile device is possible, and the current processing ability of these devices is suitable for DSP tasks.

It is still questionable whether the topographic imaging currently possible with mobile devices will be adequate for medical diagnosis. However, this review has shown that autonomous diagnosis of KC from topographic images is achievable, and it is feasible that a mobile device is used for processing results.

Chapter 3

Methodology and Design

3.1 Chapter Overview

This chapter describes the methodology, design and planning used for this project. The methodology is summarised in the next section and then elaborated on within this chapter.

For the design and planning of this project, the aims are re-stated, the objectives expanded on, and the scope and limitations are defined. The design and planning included determining the required resources, setting critical tasks, creating a schedule, and performing a risk assessment. All these areas are discussed further.

The data used for this project was from the Australian Study of Keratoconus (ASK). As this data was patient images, there was an ethical approval process to receive access. The ethical approval process and available data are discussed further within this chapter.

3.2 Project Development

The aims and objectives were specified to begin project development. Once this was established, it was essential to determine the scope and limitations to ensure the project was well defined.

3.2.1 Aims and Objectives

The overall projects aims and objectives were in Chapter 1. However, to reinforce the essential aspects, some will be repeated and expanded on here.

The project aim is to determine if an ANN and a mobile device can be used to help diagnose corneal abnormalities.

The hypothesis is that an image taken from a mobile device can be reconstructed into a corneal topography image and can be utilised for automatic diagnosis of abnormalities.

Diagnosis would be without the need for an extensive review from the ophthalmologist or specialised imaging equipment. The conditions and abnormalities that can affect the cornea are broad. As such, this project focussed on the common corneal disease KC. This project will aim to determine if an automatic diagnosis of KC is feasible using an image from a Pentacam and investigate adapting this for a mobile device.

For this system to be feasible, it needs to show it can accurately identify KC from a corneal topographic image. The objectives below are an extension from the ones stated in Chapter 1:

(i) Obtain patient images of corneas with and without KC.

(ii) Adapt different ANNs for the project.

(iii) Create algorithms for standardising and preprocessing of all images.

(iv) Test compatibility of the ANN and preprocessing algorithms.

(v) Determine the best network, the ideal parameters, and ideal training data.

(vi) Carry out extensive testing and determine networks performance.

(vii) Test independent data on trained ANNs.

(viii) Investigate reconstructing corneal topographic images from a mobile device.

(ix) Create Android® mobile application for demonstration.

3.2.2 Scope and Limitations

The scope of this project will be limited to a desktop computer, Android device, and detecting KC from Pentacam images. However, if successful, this project will offer a proof of concept for numerous corneal abnormality detection.

The most significant limitation of this project will be time and data, secondary to that will be budget. The time constraint will limit the amount of testing and refining that can be trialled throughout this project. The scope of results is confined by the amount and quality of data that is provided by ASK. Furthermore, as it will be completed externally to the University and with no financial support, the project was limited to hardware and software that is already on hand or is of a relatively low fiscal burden.

While the methodology and design were successful for this specific data, to expand on this in the future, it will need to be re-visited. As more diverse data is made available the preprocessing and ANN techniques may need to be more flexible and further developed.

3.3 Methodology Outline

While the methodology is at length in this chapter, a brief overview is shown here. The paper determines if ANNs can be used to differentiate between patients with KC and patients without KC. Furthermore, it was discovered whether or not a mobile device can process images to create a prediction from the final model.

An overall quantitative methodological approach was used, concentrating on data collection and analysis. The project uses images from a Pentcam supplied by ASK and as such results are within the scope of the specific data that was available. However, this does still provide proof of the hypothesis.

The method employed is summarised below.

- Data
  - Four map topographic images of patients from a Pentacam.
  - Images included patients diagnosed with KC and patients with other diagnoses.

  - Separated into training and test data.

- Image Preprocessing
  - Superfluous information removed from images and separated into their four maps.
  - The aspect ratio and the size of the images made uniform.
  - Images were scaled. The final model scaled all images to 10% of their original size.
  - All images separated into the individual RGB channels and greyscale.
  - Data augmentation was implemented to increase the quantity of training data.
  - Images converted to a single-column matrix and all inputs normalised.

- Artificial Neural Networks
  - Five ANNs used for the complete model.
  - A fully connected ANN architecture was used for all.
  - Supervised learning techniques were used.
  - Activation functions sigmoid, and swish both trialled. The final model employed sigmoid.
  - Three backpropagation modes were trialled. The final model used the online mode.
  - Learning rate depending on the network was either 0.001 or 0.05 for training the final model.
  - The momentum term for weight updating was left constant.
  - Xaviers Initialisation Technique was used for initialising the weights.
  - The bias's were initialised to zero for the final model.

- Preliminary Training
  - Networks trained on all single RGB channels to determine convergence times.
  - Each topographic map was trained independently.

- Dimension Testing
  - Test to find the best dimensions to make the data invariant too.
  - To use data efficiently, the leave-one-out cross-validation technique was used.

  - The process was automated and took approximately 160 hours for all maps.
  - Average certainty results were used to determine the dimension for the final model.

- Final Model
  - Multiple ANNs used to divide computational efforts into manageable parts.
  - Results used from dimension testing were used for training the four maps ANNs.
  - Networks linked using the fifth ANN.

- Testing and Adjustments
  - The final model was tested on data not used for training.
  - This testing was used to determine improvements in the current model.
  - Adjustments made to the final model based on testing.

- Results and Analyse
  - The percentage that the final model differentiates between KC and control was used as results.
  - The model's certainty for all results was analysed to determine significance.
  - The images that the model could not predict are highlighted.
  - Results can be generalised to offer suitability of applying these techniques for other corneal abnormalities.

- Android Application
  - Developed using Android Studio.
  - The final model classifier is used to produce mobile app results.
  - The application was to support the theory a mobile can process an image once the model is trained.
  - Future development ideas are discussed based on found difficulties.

This methodology for this project was acceptable for this specific dataset and project aims. The processing techniques were kept as simple as practical while still producing

decent results and a proof of concept. However, the preprocessing and ANN techniques may need to be revisited if more diverse data is used in the future.

The Android application was suitable to support the theory that a mobile device could aid in KC diagnosis, although there are some improvements to the fundamental design that can be made to make it more efficient.

3.4 Data Collection

The availability of data for this project is critical. The data is vital to ensure the networks are both trained and tested efficiently. To ensure the models are trained adequately, there was a need for a robust database of patients with KC and just as many patients without KC as the control.

Throughout the risk assessment process in Section 3.14.3, the data collection was the area of most significant risk. This section will describe the data provided by ASK, highlight procurement of data, limitation of the data, confidentiality, ethics, and security.

3.4.1 Description

The data for this project was four map topographic images from ASK's Oculus Pentacam. Figure 2.4 shows a similar example of available data. This data included images of patients with and without KC. As patients are not referred to the Pentacam unless their ophthalmologist already suspects a corneal abnormality or disease, there is minimal data available of healthy eyes. Instead, most control images used are of patients with other corneal or refractive conditions to KC. Healthy eyes compared to eyes with KC would have been simpler to differentiate between; this added challenge poses an additional proof of concept.

The KC data initially made available included 51 patient images. Of those 51, the KC diagnosis of each patient included 17 sub-clinical, 9 mild, and 25 severe. The control images initially made available included 35 patient images. The patient diagnosis for these control images included glaucoma, cataracts, lens subluxation, trauma, post Lasik, and one healthy cornea. This first dataset was used for preliminary training and dimensionality

testing. The training and testing are discussed further in Section 3.7 and 3.8.

The last dataset made available included 45 control images and 49 KC images. The diagnosis of the patients in this set of control images was unknown. Of the 49 KC images, the patient diagnosis of KC includes 23 mild and 26 moderate.

There were images of both the left and right eyes in all data. However, as each patients eye was diagnosed on its merits with no comparison to the other eye, this is not of significance to this project.

Techniques were used within this project to make the most of the available data. Each image was divided into its four topographic maps, and a method called data augmentation was executed. These techniques essentially enabled the information for training to be increased by a factor of 16. This is further discussed in Section 3.5 and Chapter 4.

3.4.2 Procurement

Obtaining the necessary data for this project proved to be a challenging task. The first allocation of data was not made available until the second half of this project and the final distribution after that. There were several contributing factors to the delay and limited data availability.

As the data was patient images, there were ethical considerations that had to be considered. This ethical approval process was quite extensive, and this was granted approximately halfway through the project. Ethics approvals are shown by Appendix C and the required procedure is discussed in Section 3.13. Additionally, obtaining images from the Pentacam was an arduous manual task.

The Pentacam had the data stored within, and it is still operational medical equipment. Hence, access to download the images was only made available after hours where it would not affect normal operations. Furthermore, patient records had to be manually checked to find patients with KC and then cross-referenced with patient identification on the Pentacam. Each patient image was then downloaded manually one at a time. This was the same process to find patient data without KC.

These factors all contributed to the delay and the quantity of data that was made available

for this research.

3.4.3 Confidentiality and Security

All patient names were removed from images to ensure confidentiality. The only distinguishable feature left was the patient ID. Hence, a patient is only identifiable if used in conjunction with patient records.

This removal of patient names simplified the process to obtain the images. The data was transferred securely over the internet. However, there is still a requirement to ensure the security of the information. This data security was addressed within the risk assessment, and all information is only to be used for its intended purpose and not re-distributed.

3.5 Image Data Preprocessing

As was found by the literature review, ANNs for image recognition rely on the quality of the dataset. Some important considerations that this project addressed for image data preparation are below (MissingLink 2019). Most of these parameters and considerations have been discussed in Chapter 2. However, the specific methodology for each is stated.

Image size - Higher quality images will give more information but require more ANN nodes and computation. This project explored the compromise between more information and available computational effort.

The number of images - As a general rule, the more data that is available for a model, the more accurate it is. The restriction for the research was the number of images made available by ASK.

The number of channels - Greyscale images have two channels, and colour images typically have three channels. This project utilised colour images and performed extensive testing on dimensionality reduction. This enabled two-dimensional images as model inputs.

Aspect ratio - All images must have the same size and aspect ratio. As literature suggested, this project ensured all images were square.

Image scaling - Concluding making images square, they were scaled to fit each ANN model.

Normalising inputs - This ensures all pixels have a uniform distribution to ensure convergence is faster during training network. The inputs were normalised by subtracting the mean from each pixel and dividing the result by the standard deviation.

Dimensionality reduction - Reducing dimensions can be done to make the ANN invariant to that dimension or to make training less computationally intensive. This report did extensive testing on reducing RGB images to individual channels and greyscale.

Data Augmentation - This involves augmenting existing data with perturbed variations of current images. Augmentation is done to expose the ANN to a wide variety, so it is less likely to identify unwanted characteristics. Data augmentation was achieved by scaling and rotating all available images.

The preprocessing techniques for all of the data are implemented using MATLAB. For efficiency, during the ANNs learning process, all preprocessing is completed and saved prior. The algorithms used to perform this preprocessing is discussed in Chapter 4.

This project adopted the philosophy of keeping the preprocessing techniques as simple as practical while still producing good results. There are vast amounts of preprocessing techniques available. However, through research in the literature review and experimentation, the methods used for this project were suitable for this particular dataset and methodology.

3.6 Artificial Neural Networks

After training images are prepared, an ANN processed them and made predictions on new, unknown images. This project used five traditional ANNs to create the final model, and a supervised learning methodology was adopted. The traditional ANN model uses a fully connected architecture, as shown in Figure 1.2.

The final model for this projected consisted of five networks because each Pentacam image was separated into four maps, before being linked using an additional network. The ANNs used were all developed in MATLAB. To enable full flexibility, these were designed from

the ground up without the use of any toolboxes. This methodology allowed the model to be created specifically for the data and changes to parameters and techniques made efficiently.

This section will explain the specifics about the ANNs for this project. Additionally, the concepts and specific techniques applied to the complete model will be described.

3.6.1 Complete Network Summary

Every neuron in the input layer takes one pixel of the image and carry out the computation. That result is passed on to additional neural layers until it gets to the end of the process, and the network has generated a prediction. The project uses a non-linear activation function to create the prediction. This process is repeated for a large number of images, and the network will learn the most suitable weights and bias for each neuron. This learning process is called backpropagation (MissingLink 2019). The activation functions, backpropagation, mathematical functions, and initialisation techniques used for this research are described in Sections 3.6.2, 3.6.3, 3.6.4, and 3.6.5.

3.6.2 Activation Functions

The forward propagation carried out by each neuron can be summarised as, taking the input and multiplying it by the weight of the neuron, adding a bias, feeding to the activation function, and finally, the result is transmitted to the next layer. This activation function can be a binary step, linear, and non-linear functions.

A binary step and linear activation function were not suitable for this project. They were not ideal because a binary step does not allow multi-value outputs and a linear function within an ANN is simply a linear regression model. Furthermore, they both do not accommodate backpropagation. However, a non-linear activation function alleviates these issues and allows the model to create complex mappings between input and output, which is essential for learning and modelling images. The use of a non-linear activation function enables the introduction of real-world properties to ANNs.

There are many non-linear activation functions used for ANNs, each with their advantages and disadvantages. This project utilised a sigmoid activation function and then expanded

to trial a self-gated activation function called swish (Ramachandran, Zoph & Le 2017*a*). Sigmoid is a novel activation function for smaller ANNs. However, as will be explained, as the ANN gets more in-depth, a sigmoid function may become inefficient where a swish function will not. Due to this phenomena, the project began with sigmoid functions but as the number of neurons and layers are increased the swish activation function was trialled. The sigmoid function is shown by Equation 3.1 and Figure 3.1a.

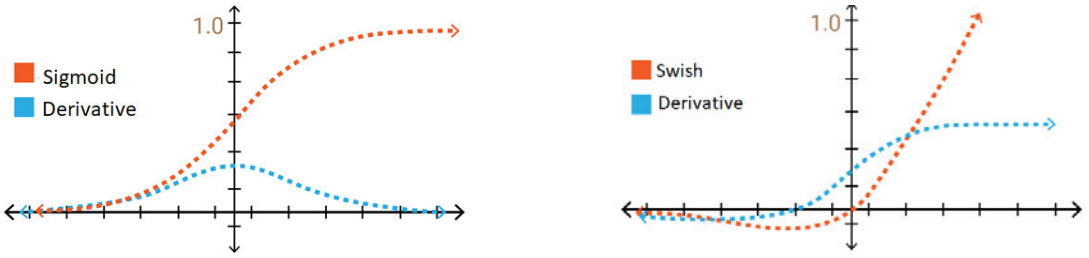
$$S(x) = \frac{1}{1 + e^{-x}} \tag{3.1}$$

As can be seen, the use of a sigmoid function normalises the output between 0 and 1 when approaching negative and positive infinity. This trait of sigmoid functions ensures that the activation value does not vanish. However, by inspecting Figure 3.1a towards the end of the function on the x-axis, the derivative values become very small and approach zero. These phenomena are termed the vanishing gradient effect and learning is minimal at these points. Learning is minimal as backpropagation relies on the derivative to determine corrections (Kızrak 2019). This can cause the network to learn inefficiently or attach to a local solution.

A solution to the vanishing gradient problem is the swish activation function. This function is x multiplied by sigmoid and is shown by Equation 3.2 and Figure 3.1b.

$$\sigma(x) = \frac{x}{1 + e^{-x}} \tag{3.2}$$

In an ANN with many neurons, a sigmoid function will cause a significant number of neurons to be activated the same way. Hence, it is computationally intensive. Swish is a function where fewer neurons are activated in the same way than sigmoid, and due to there not being a zero-value region, learning occurs in all areas. Swish is a good compromise between many neurons activated and parts that learning will occur. As the project utilises deeper ANNs being able to train the model efficiently will be critical, and swish was explored for this.



(a) Sigmoid activation function and derivative.

(b) Swish activation function and derivative.

Figure 3.1: Non-linear activation functions (adapted from Kızrak (2019)).

3.6.3 Backpropagation

Backpropagation is the algorithm used in this project to train the ANNs. It adjusts the weights and biases of the neurons so that the output becomes closer to the known exact result. The first stages of learning are initialisation of the data and forward propagation. For learning to occur the next steps are: error function, backpropagation with weight and bias update, and iterate until convergence.

The error function captures the delta between the correct output and the actual output of the model given its current weights and biases. The error formula used was the squared error function, and it is defined by Equation 3.5. Backpropagation is the process of using gradient descent to change the weights and biases to bring the error function to a minimum. Figure 3.2 shows a simple example of how and by how much a network would make its adjustments based on the error.

As the weights and biases are updated at a small delta step, many iterations are required to converge to an optimum solution. The amount of repetitions is dependent on the learning rate. This learning rate was varied from 0.001 to 0.25 to determine the correct fit for each model.

At the end of this learning process, the model is ready to make predictions on new unknown data. For accurate results, the network was tested on data not used for training. The next section will summarise the mathematics this research required for forward and back propoagation.

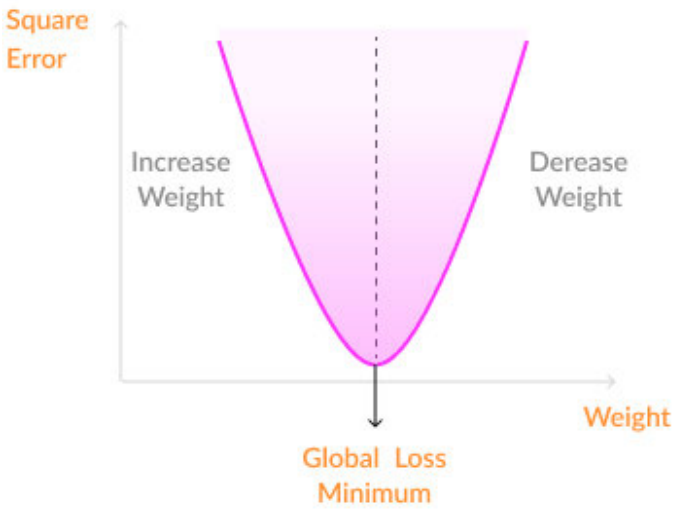


Figure 3.2: Backpropagation curve. This shows a simple example of when and by how much a network will increase or decrease a neurons weight (adapted from MissingLink (2019)).

3.6.4 Mathematics and Concepts

This section will highlight the maths and concepts that were needed for forward and backpropagation. This section is developed from research during the literature review in Chapter 2, and the structure and terminology are commonly accepted for ANN explanations (Joglekar 2017).

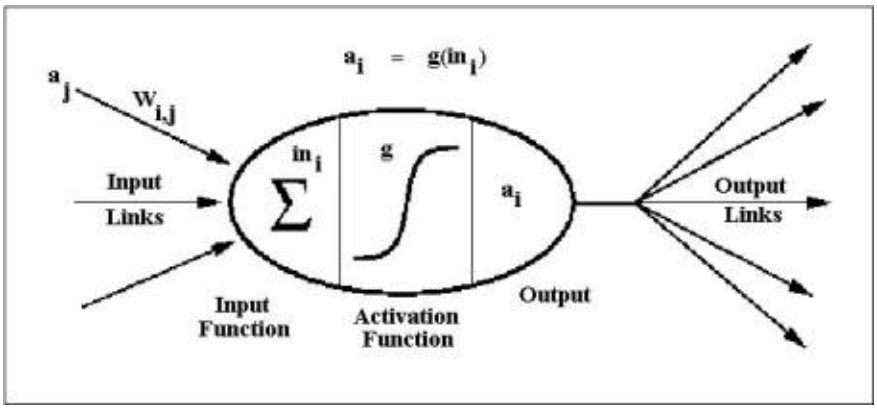


Figure 3.3: Workings of a single neuron (N_i). Which represents its output activation as a_i determined from its inputs, N_j being one of those inputs (adapted from Joglekar (2017)).

Figure 3.3 shows the working of a single neuron within an ANN. The nomenclature from this figure will be used for Equations 3.3 to 3.13.

N_i has a weight assigned for each of its inputs. For example in Figure 3.3 the input N_j has a weight assigned of $w_{i,j}$. The first step in the forward propagation is that N_i calculates

a weighted sum of all of its inputs. This is shown by Equation 3.3, where $Inputs(N_i)$ is all neurons that provide input to N_i.

$$in_i = \sum_{k \in Inputs(N_i)} w_{i,k} a_k \tag{3.3}$$

This sum (in_i) is then passed to the activation function, shown by Equation 3.4.

$$a_i = g(in_i) \tag{3.4}$$

As the activation function for this project will be the same for all neurons, it can be represented by g instead of g_i. The activation functions used are as previously discussed and are shown in Equation 3.1 and 3.2. After these calculations have been performed, the training process needs to ensure each weight ($w_{i,j}$) is right, so the overall network gives an accurate prediction. This is where backpropagation is used to optimise all of the weights. Specifically, the backpropagation is optimising the error function.

The mean-squared-error (MSE) function was used to determine the error for this project. The output of this single neuron is N_o, and for a given image sample, the expected output will be t_o. As N_o gives an output equal to its activation value shown by a_o. Then the MSE error can be shown by Equation 3.5.

$$E = \frac{1}{2}(t_o - a_o)^2 \tag{3.5}$$

As a_o is dependent on the weights are given by N_o to its inputs. Then E is also reliant on these weights. As every training sample will result in a different MSE value, the goal of backpropagation is to minimise the cost of E for all training data as an average. Cost minimisation is achieved by adjusting the weights $w_{i,j}$ using a technique called gradient descent.

The gradient of the function with respects to its variables determines how much a function will change with a unit increase/decrease of its variables. This is obtained by differentiating the function concerning its variable, and because the functions in this project are dependent on a variety of parameters, the partial derivative is used. Figure 3.4 is a visual

representation of how gradient descent can find one of the solutions.

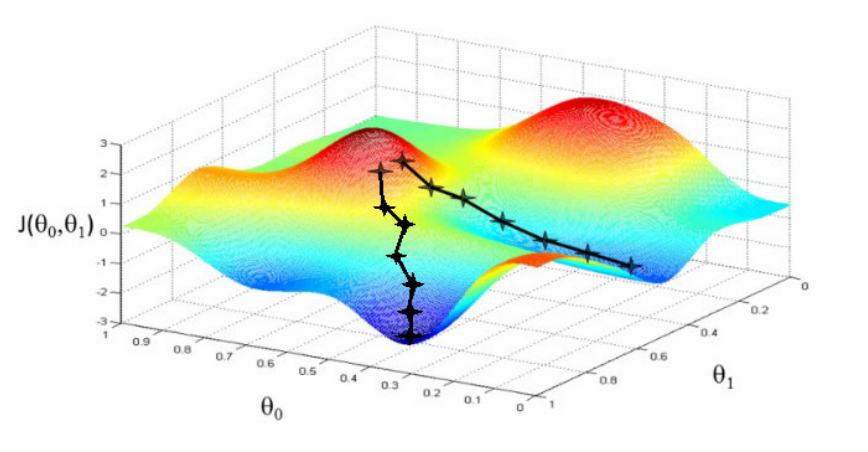


Figure 3.4: Gradient descent visual example. The model estimates the weights in many iterations by reducing the error. This shows how two different training attempts can result in an overall similar final solution but at different positions (adapted from Vryniotis (2013)).

The red region in Figure 3.4 represents higher function value, while the blue regions lower function value. The model is attempting to reach the lowest point. The model takes small incremental steps where the terrain downwards slope is at a maximum. Hence, the direction is the gradient of the height with respects to the current location (x, y). In this example, the overall direction to move is dependent on the gradients for x and y. Which looks like $\frac{\partial height}{\partial x}$ and $\frac{\partial height}{\partial y}$.

In regards to this project and ANN training, backpropagation is used to reduce the MSE function E by utilising the gradient to individual weights, $\frac{\partial E}{\partial w_{i,j}}$. Once the network has carried out a forward pass, the networks outputs in conjunction with the known correct value were used to calculate the MSE using Equation 3.5. The weights are then updated using the gradients. This updating begins at the outer most weights and works its way back to the lower layers. The rationale behind this is that the external most errors are known first, and the central neurons errors and gradient are derived from them.

As shown the gradient is calculated using $\frac{\partial E}{\partial w_{i,j}}$ for a given sample, where $w_{i,j}$ is the weight given by N_i to N_j. Using the chain rule, Equation 3.6 can be determined.

$$\frac{\partial E}{\partial w_{i,j}} = \frac{\partial E}{\partial in_i} \frac{\partial in_i}{\partial w_{i,j}} \tag{3.6}$$

With Equation 3.3, Equation 3.4, and Equation 3.6 in mind and utilising differentiation

and the chain rule again. Equation 3.7 can be found.

$$\frac{\partial E}{\partial w_{i,j}} = a_j \frac{\partial E}{\partial in_i} \tag{3.7}$$

From the definition of the activation function and current weights it is known that a_j is equivalent to $\frac{\partial a_i}{\partial in_i}$. The sigmoid activation function was advantageous to use as it's derivative concerning its input is an uncomplicated expression. This is shown by Equation 3.8.

$$\frac{dg}{dx} = g(x)(1 - g(x)) \tag{3.8}$$

While the swish activation functions derivative is slightly more complex than sigmoid, it is still relatively simple. It's derivative is shown by Equation 3.9 where $S(x)$ is the sigmoid activation function.

$$\frac{dg}{dx} = g(x) + S(x)(1 - g(x)) \tag{3.9}$$

The increase in complexity can be necessary for training efficiencies as an ANN gets deeper. The rest of the ANN explanation will be shown using the sigmoid activation function and its derivative. However, it is directly transferrable to the swish activation function and its derivative.

Combing the information from Equation 3.7 and Equation 3.8 the gradient can be represented by Equation 3.10.

$$\frac{\partial E}{\partial w_{i,j}} = a_j a_i (1 - a_i) \frac{\partial E}{\partial a_i} \tag{3.10}$$

Observing the last term of Equation 3.10 it can be seen that N_i cannot be an input neuron as it is receiving input from another neuron. Therefore, it must be an inner neuron or an output neuron. With this in mind, the last term can be simplified to Equation 3.11.

$$\frac{\partial E}{\partial a_i} = \sum_k w_{k,i} a_k (1 - a_k) \frac{\partial E}{\partial a_k} \tag{3.11}$$

It may not seem like a simplification. However, the backpropagation term $\frac{\partial E}{\partial a_k}$ is known from previous calculations.

Once the gradient was found from Equation 3.10 and 3.11 the weights were then updated. This project used Equation 3.12 to change the weight for a given training sample at time instance t.

$$\Delta w_{i,j}^{t} = -\epsilon \frac{\partial E}{\partial w_{i,j}} + \alpha \Delta w_{i,j}^{t-1} \tag{3.12}$$

The value ϵ is the learning rate, and it is negative, so the progress is towards negating the gradient. The learning should not change drastically based on one training sample. Hence, the learning rate was always below 0.25 for training. Although, throughout this project, the learning rate was varied to as low as 0.001. This experimentation was to ensure the learning rate was not too high, causing oscillations while searching for a solution or too small, causing the network to stagnate during training.

In Equation 3.12 the term α is called the momentum. This momentum term is to help prevent the network from finding a local optimum solution in contrast to the global optimum solution. This term causes the current weight update to be dependent on the last weight update. Therefore, the update will not vary drastically. If the solution were the global optimum, the search would make its way back to the point as the momentum will continually reduce. The literature review in Section 2.7 did show a case where differing momentum was beneficial. However, it was left constant for this research as careful initialisation techniques and network complexities were adequate to prevent the network from finding local solutions.

It has been shown how forward and backpropagation are processed. Additionally, it is necessary to explain how the weights were initialised and how often they were updated. This is discussed in Section 3.6.5.

3.6.5 Weight Update and Initialisation

It has been shown how the weights and biases will be updated. However, it is essential to determine the frequency and initialisation of all of these within the network. There are three modes that this project looked at for the rate of updating the weights. These

modes include online, batch, and stochastic (Joglekar 2017).

This project began its initial training utilising the online mode. In this mode, the weights were updated after every training sample. This means, the forward and backpropagation occur on every training sample, and this was initially accepted as the initial training data was relatively small. As the data got more substantial, and there were suspected outliers, it was necessary to try other modes to achieve better results.

Batch mode is a technique where weights are updated after accumulating gradients concerning all samples in the training set. In other terms, backpropagation will only occur once per epoch. This was a useful technique of training data due to suspected outliers as it is more resistant to variance than the online mode. Unfortunately, even with the swish activation function, this increased the training time substantially and did not produce adequate improvements.

The stochastic method, also known as mini-batch adds pseudo randomisation to the frequency of weight updating. The complete training set is randomly divided into smaller batches, and backward-propagation is performed after accumulating gradients for all of the samples within the mini-batch. Stochastic mode offers a good compromise between the online and batch modes. Furthermore, as the random mini-batches will be determined after each epoch, the chances of finding a local optimum are reduced. The stochastic method was implemented; however, as with the batch mode, it did not show any improvement in comparison to the added complexity.

The online mode ensued good results when additional data was available for training and outliers did not appear to affect training accuracy. Furthermore, reduced training time was necessary due to available hardware during testing. This is further discussed in Chapter 4 and 5.

Lastly, it was necessary to define how to initialise all weights and bias. The initialisation was imperative. If initialisation values are too small, the signal shrinks as it passes through the network until it is tiny and not useful. Conversely, if the weights are too large, the signal increases as it passes through each layer until it is so large it isn't helpful.

A novel solution discussed extensively in literature is the Xavier Initialisation Technique. While there are many variants of this technique, this was a more than adequate solution. Instead of just randomly allocating values to weights which can lead to activation functions

starting in dead or saturated regions, this technique bounded the random distribution by previous and current layer neurons. This maintains a similar variance of activations and back-propagated gradients throughout all layers (Dellinger 2018).

The weight for each neuron was assigned by finding a pseudo-random number with a normal distribution with a mean of zero and a standard deviation defined by Equation 3.13.

$$\sigma = \sqrt{\frac{2}{\sum N_i + \sum N_o}} \tag{3.13}$$

Where $\sum N_i$ is the number of input neurons to the weight tensor and $\sum N_o$ is the number of output neurons in the weight tensor. As discussed, there must be careful considerations when initialising weights. But, there are fewer considerations for the bias. There are fewer considerations because the gradients concerning the bias depend solely on the linear activation of that layer, and not the gradients of deeper layers. Hence, there is no vanishing or exploding gradient for the bias terms, and it can be initialised to zero or tiny terms (Doshi 2018). This project tried initialising the bias to small non-zero values and zero.

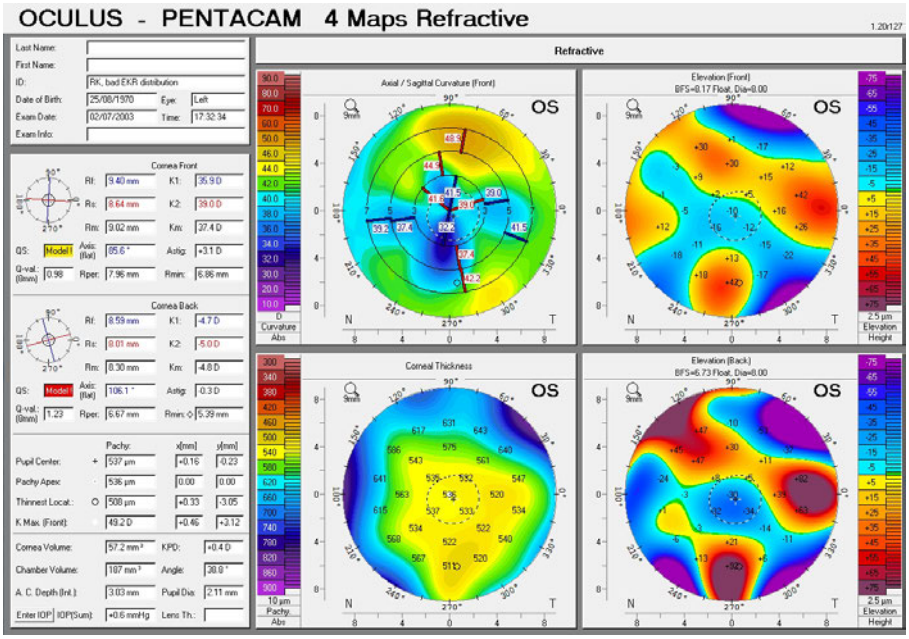
3.7 Preliminary Training

The depth of a traditional colour image is three as it has a dimension for each of the RGB shades. This project divided each image into these three dimensions to determine if the ANN can learn on a single dimension.

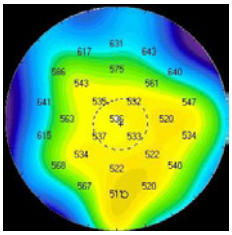
Preliminary training was performed on each of the individual topographic maps to ascertain which of the RGB dimensions the network can be made invariant to and still converge. Each of the four maps per image was trained independently to each other.

Making each image invariant to two of the three RGB channels was deemed necessary to reduce computational effort while training the ANN.

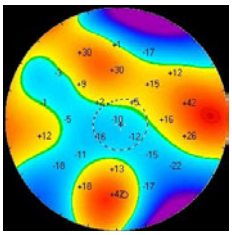
Figure 3.5 highlights the before and after preprocessing for the individual topographic maps. The outputs in Figure 3.5b and 3.5c are examples of different topographic maps that are inputs for the model once further preprocessing steps were completed.



(a) Pentacam four maps refractive image with patient data removed.



(b) Corneal thickness.



(c) Elevation front.

Figure 3.5: Pentacam four maps image and example output of preprocessing to be used for model inputs.

The specific parameters and complexity of the system for this training are in Chapter 4. Observing that the network would converge on a particular dimension did not supply adequate information to determine which dimension could be used for the final model. Further testing was required to resolve this lack of knowledge, discussed in Section 3.8.

3.8 Dimension Testing

Dimension testing is an extension of the preliminary training to gather more reliable results for the final model. This testing involved training ANNs on topographic maps that were reduced to a single RGB dimension and determining how it performed on unique data not used for training.

For the most successful final model, each of the individual topographic maps was tested independently. These independent results are used to create the final model. As discussed, the original data provided was quite limited. However, to ensure the available data is managed efficiently, training and testing were done using the leave-one-out cross-validation technique (Kohavi et al. 1995). This technique is discussed further in Chapter 5.

However, it is essentially removing an image one at a time to be used as independent data for testing. This testing of single images was automatically repeated for all data.

The networks average certainty on each RGB channel was used to select what single dimension is used for each map when developing the final model. This testing was crucial to the project as literature highlights the importance of testing a network on unique data. This testing required lengthy processing time of approximately 160 hours. However, it indicated a solid proof of method and promoted a robust final model. The specifics and results from this validation are in Chapter 5.

3.9 Final Model

The final model consists of five ANNs. The decision to use five ANNs was to ensure the training of the final model was divided into computational components that are manageable.

The results from the dimension testing were used to create the final model. Each topographic map during the dimension testing was invariant to two of the three RGB channels. The channel that produced the most accurate results is used for training the final model.

This final ANN receives inputs from all topographic maps available on each Pentacam image. Each topographic maps forward propagation prediction is supplied as the inputs for this last network. This final model is then trained using the labelled training data. This training is further shown in Chapter 4.

Once the final model was trained, it was tested on data not previously seen by the current model.

3.10 Testing and Adjustments

For meaningful testing, all the literature stressed the need for an ANN to be tested on data not used for training the classifier. As such, once the final model was prepared, it was tested on new images. The model can present three possible results. These results included a correct, an incorrect, or an uncertain prediction. With every prediction, the

model will give a certainty figure, if this value is below the set prediction tolerance an uncertain prediction is then the output.

This testing was used to detect any weaknesses in the final model. Based on the results during this stage, adjustments to the model's parameters were made. This testing and adjustment process was repeated several times. The changes made during this process included increasing the number of neurons, changing the learning rate, varying backpropagation modes, models prediction tolerance, and increasing training data.

This is further discussed in Chapter 4 and 5.

3.11 Results and Analysis

The results are recorded as a percentage that the final model can differentiate between the images of patients with KC and the images of patients without KC. The networks certainty of each prediction was examined. The analysis was to determine if there was any correlation between confidence and KC severity. Additionally, the data that the model did not predict correctly was further scrutinised to learn why.

This analysis of results showed a suitable proof of concept which could be generalised for other corneal abnormalities.

3.12 Android Application

An Android application was developed to aid in providing further proof of the original hypothesis. The app was developed in Android studio and implemented on several devices. It was designed to work on any Android device with an operating system at or above Android version 7.0.0. For this project, the app was installed on a Samsung® Galaxy S7 and a Samsung Galaxy Tab E.

The app was designed to use the results from the classifier trained for the final model. The app shows preprocessing steps to the user and then provide the prediction, certainty, and actual ophthalmologist diagnosis on the last screen. The user selects an uploaded Pentacam image. The app will then step through three screens showing the user the

outputs after the first preprocessing step, then after dimensionality reduction, and finally, the results and initial user selection. The images uploaded to the device was data not used for training the classifier.

The app does have limited functionality and needs to be made more efficient in the future. However, it was designed to highlight that a mobile app could be created for analysing this type of data. Furthermore, the screens that show steps of preprocessing would not be required for the final design. It was merely to give a visual explanation of some parts of this project.

This mobile app and functionality is further explained in Chapter 6 and 7.

3.13 Ethics

As this project required access to patient data and clinical records, there are ethical concerns to take into consideration. To ensure that this research project and patients are protected; there was a requirement to apply for Honorary Researcher with the Royal Victorian Eye and Ear Hospital. This application is with the Human Research Ethics Committee, which is established per the National Health and Medical Research Council (NHMRC) and requirements to the National Statement on Ethical Conduct in Human Research (2015).

The objective of applying as an Honorary Researcher is to ensure that all research conducted complies to the principles of the NHMRC. Furthermore, if ethical issues arise during this project, they are administered by a properly constituted committee (Committees 2014). For researches that do not have an eye and ear appointment and require access to medical records, the hospital can grant Honorary Research appointment. This project had to submit and complete:

- Honorary Researcher Application
- Police check
- Health Record Act training

By undertaking adequate training and an extensive application procedure, it ensured that

this research treated all personal data with due ethical concerns. This ensures that personal information is kept confidential and that this project does not come under scrutiny. The application explained has been submitted and approved. The ethics approval has been granted and shown at Appendix C.

3.13.1 Honorary Researcher Application and Declaration

The Honorary Researcher Application is compulsory, for all new researchers who do not have an eye and ear appointment and require access to medical records. This application requires the researcher to:

- Keep all information and data confidential of individuals involved in research and to not make a copy of participants records.
- Keep all information and data confidential concerning persons or events that come to attention during research.
- Only use data collected for the study for which approval has been granted.
- Maintain security procedures for the protection of privacy.
- Comply with the Eye and Ear Hospital's Researcher Handbook (Committees 2019).

This application is at Appendix E and was submitted to The Royal Victorian Eye and Ear Hospital with supporting documentation showing a recent National Police Check, proof of Health Record Act Training, and a current Curriculum Vitae.

3.13.2 Police Check

As supporting documentation for the Honorary Researcher Application, a current Name Only National Police Check was required. The requirement was this check to be within the last three months. This process involved supplying a personal history online, paying a small fee, and verifying identity at a local New South Wales Police Station.

This process took approximately ten days, and a copy of the police check was attached as supporting documentation to the Honorary Researcher Application.

3.13.3 Health Record Act Training

The Health Records Act Training is an extensive course developed from the Health Complaints Commissioner (HCC) to give researchers an understanding of the Health Records Act (2001). This training taught the following information on:

- relevant health information
- the Health Records Act
- the Health Privacy Principles (HPPs)
- access to health information
- exemptions to the Health Records Act
- dealing with complaints, and
- offences and breaches.

To complete this course, the requirement was to complete quizzes and an assessment piece. This course gave a fundamental understanding of handling health records. Additionally, it ensures the project will abide by the eleven Health Privacy Principles and complies with all associated legalisation. The proof of training certificate provided to The Royal Victorian Eye and Ear Hospital is shown by Appendix F

3.14 Project Planning

To ensure the success of this project and it's objectives, it was necessary to have a solid plan. For the preparation of this project, it was divided into key areas, including the resource requirements, essential tasks, project schedule, and risk assessment.

3.14.1 Resource Requirements

Resources were required to ensure this project could meet its objectives. The primary and most significant is time. This project also needed other resources, including specific software and hardware. Most of these resources were already accessible. If they were not available, they were acquired before each phase of work.

In regards to time, this project is estimated to have been more than 500 hours of work. This time was spread across the project duration and averaged out to a consistent 10 – 15 hours of work each week. This time was utilised by continually investigating recent developments in literature, planning, ethics approvals, obtaining required data, adapting algorithms, optimising and testing, analysing results, communication with USQ, contact with ASK, writing the paper and continually reviewing. This contribution of time was pivotal to the success of the project.

With respects to ethical approvals, the research used current patients data via the involvement of ASK. This project was always transparent with ASK and only used the data in compliance with approvals. There were contingency plans in place if permissions for data on current patients were not approved or if there was limited availability. However, these contingencies were not needs as data was made available. Data availability is addressed further within the risk assessment. The resource requirements are shown in Table 3.1. These resource requirements were developed before commencing the project; as such, some resources listed were not needed. The comments in Table 3.1 discuss resources that were not required.

Table 3.1: Project resource requirements.

Item	Quantity	Source	Cost	Comment
Laptop with Windows	1	Student	In possession	Word processing and low-level testing.
PC with Acceptable RAM and GPU	1	Student	$1889.00	It is necessary to have PC with stand-alone GPU for data processing.
MATLAB	1	Student	In possession	Required for the majority of programming.
MATLAB Deep Learning Toolbox	1	MathWorks	TBA	It was determined this was not necessary after testing this projects initial ANN.
Microsoft Word	1	Student	In possession	Word processing of dissertation.
Microsoft Excel	1	Student	In possession	Data analysis and presentation.
Microsoft PowerPoint	1	Student	In possession	Project presentation and poster.
National Police Check (Name Only)	N/A	New South Wales Police Force	$57.20	Required for Honorary Researcher Application.
KC Images	As many as available	ASK	Nil	It is imperative that there are many images to train the network.
Non-KC as Control	As many as possible	ASK	Nil	Same as above.
Fuch's Dystrophy Images	As many as possible	ASK	Nil	These images were not made available.

Table 3.1 shows the PC with a standalone GPU is a high cost. However, having a high-performance PC with a graphics card saved considerable time when processing many images. The time that it saved was well worth the investment. During the dimension testing, it was calculated that the upgraded PC was over ten times quicker. This efficiency was significant as the total processing time during that stage was about 160 hours.

The cost of this project was acceptable and was approximately $2000. The fiscal commitment is not even comparative when compared to the time devotion.

3.14.2 Key Tasks and Project Schedule

To ensure the project met its objectives and milestones were monitored. The research was broken down into 7 phases:

- Phase 1 - Project preparation phase.
- Phase 2 - Data collection.
- Phase 3 - Initial testing.
- Phase 4 - Create preprocessing algorithms.
- Phase 5 - Create ANNs.
- Phase 6 – Evaluate performance.
- Phase 7 - Write-up and presentation.

Table 3.2, 3.3, and 3.4 further breaks these phases down into specific tasks. These phases of work were created before the commencement of the research. Hence, they are in future tense.

Table 3.2: Project phases 1 to 3.

Phase 1 - Project Preparation Phase	
1A	Project approval – Obtain formal approval to commence the project from USQ.
1B	Resource procurement – Ensure all resources that are required for the project have been obtained.
1C	Networking – Contact various ophthalmology resources to obtain access to shared image databases.
1D	Permissions – Apply for permission to get data from patients.
1E	Literature – Perform Literature review.
Phase 2 - Data Collection	
2A	Approvals and procurement - Obtain ethical approvals to utilise corneal topography images from the hospital database. Receive images with security of data in mind.
2B	Database - Collect KC and non-KC Pentcam images from online resources. Combine these with database from ASK.
2C	Group and sort – Discriminate images that are not appropriate and sort all data into groups. Data for training classifiers and data for testing. Each group will have KC and non-KC.
Phase 3 - Initial Testing	
3A	MATLAB training – Create an ANN and test on MINST dataset. Test and modify network to ensure it can handle changes in parameters easily.
3B	Specific training – Expand on the network from 3A to train network on similar images to the aim of the project.
3C	Document results – Determine the accuracy of this network based on the test dataset. Continually document results throughout this phase.
3D	Experimentation – Change parameters of the ANN, including the number of neurons, hidden layers, number of epochs, initialisation, and learning rate.
3E	Result review – Analyse and determine required improvements. Utilise these results as a baseline for expected outcomes.

Table 3.3: Project phases 4 to 5.

Phase 4 - Create Preprocessing Algorithms	
4A	Image cropping - Create algorithm which will automatically take four maps Pentacam image and separate and isolate map to ensure only relevant information is present.
4B	Input matrices – Convert each map to 3-dimensional matrix and isolate RGB dimensions to create 2-dimensional matrices. Additionally, create greyscale representation of each image.
4C	Input variations – Normalise inputs and size of matrices. Different inputs are used with ANN in Phase 6.
4D	Input database - Process all images and label all input databases for training in Phase 6.
Phase 5 - Create ANN	
5A	Create network – Amend network from Phase 3 to accept different input requirements from Phase 4.
5B	Train network – Do initial training on greyscale and single channel inputs from Phase 4.
5C	Document results – Document network convergence results on all dimensions and greyscale inputs.
5D	Test and optimise – Test to ensure the network is performing on database of basic input matrix. Adjust network as needed.

Table 3.4: Project phases 6 to 7.

Phase 6 – Evaluate Performance	
6A	Training – Train ANN on all different input parameters and test results on an independent dataset.
6B	Document and compare – Continually document and compare results throughout the rest of this phase.
6C	Analyse results – Continually analyse results with each network or input variation during this phase.
6D	Parameters - Change network parameter one at a time to determine the impact on results.
6E	Solution - Find most relieable solution within constraints of project.
6F	Compare – Compare optimum solution to similar literature results.
Phase 7 - Write-up and Presentation	
7A	Prepare progress report – Prepare and submit.
7B	Prepare draft dissertation – Submit draft dissertation to USQ supervisor and ASK advisor for review.
7C	Present results – A presentation of current results and findings to be prepared and shown during ENG4903.
7D	Complete – Based on feedback from supervisors the dissertation will be amended and submitted.

The tasks listed in Table 3.2, 3.3, and 3.4 were completed, and the projects aims and objectives were accomplishments. These tasks were not necessarily in the order of completion. A schedule was developed at the beginning of this research to ensure there were accountability and management of each task. This schedule indicates the required time frame for each job and is shown in Figure D.1.

Figure D.1 represents Semester 1 and 2 as weeks. Week 1 being the beginning of the project on the 25th of February 2019. The start and end time for each task are shown using a horizontal blue bar. Tasks that do not begin until the conclusion of another are dependent on these previous tasks. This project monitored these dependencies to ensure they were completed by the required week.

This planning and scheduling did help the project accomplish its aims and objectives. Although it did not always go as planned. There were several delays which were mostly

contributed to the difficulties with obtaining the data. Furthermore, an additional phase was added after the project had already commenced. This phase was to create an Android application, and due to its late addition, it was not included in the original planning.

3.14.3 Risk Assessment

A risk matrix was adapted from a typical workplace risk assessment (Fox, Spicer, Chosewood, Susi, Johns & Dotson 2018). Figure 3.6 shows the risk matrix used.

Likelihood	**Risk level**				
Almost certain	Medium	High	High	Extreme	Extreme
Likely	Medium	Medium	High	Extreme	Extreme
Possible	Low	Medium	High	High	Extreme
Unlikely	Low	Low	Medium	Medium	High
Rare	Low	Low	Medium	Medium	High
Consequence	Insignificant	Minor	Moderate	Major	Catastrophic

Figure 3.6: Risk matrix to calculate risk level (adapted from (Fox et al. 2018)).

The risk matrix, in conjunction with the USQ eight-step risk assessment process (USQ 2018), was utilised. The risk assessment assessed the risks of the dissertation completion, the security of data and personal safety. The assessment covered all relevant tasks within Table 3.2, 3.3, and 3.4. The comprehensive risk assessment for each phase was then summarised with the USQ safety risk management plan. The USQ risk assessment is shown by Figure B.1.

This assessment, identified the risk, proposed mitigation measures, and shows the risk level. The complete risk assessment is shown in Appendix B. Any final risk higher than low required special attention and careful monitoring.

Appendix B highlights that many risks required careful consideration. However, with the mitigation factors, most are assessed as low and did not need special attention with the mitigations in place.

Some risks were moderate even with additional controls. These are all related to data and are a moderate risk due to the implications they will have on the project. These were all managed prudently as their consequences to the research were significant.

3.15 Chapter Summary

This chapter has described the methodology, design, and planning of this project. The method can be summarised as utilising ANNs to differentiate between images of patients with KC to patients without KC. The methodology applies to the specific dataset that was available. This particular dataset is also a limiting factor for the project. However, it is still appropriate as a proof of concept.

The design of the final model and relevant techniques used throughout this project have been explained. The explanation included methods that were trialled throughout the process but not implemented for specific reasons.

The planning for this project began before the official commencement of the research. This included developing phases of work, creating a project plan, performing a deliberate risk assessment, and obtaining ethical approvals to access patient data. The availability of data was pivotal to success, and this was highlighted in several areas during the risk assessment.

Chapter 4

Detection Algorithms using Machine Learning Approaches

4.1 Chapter Overview

This chapter explains all the algorithms that were required to fulfil this project and details the preliminary training that was performed by these algorithms. The algorithms implement all the preprocessing and ANN techniques described in the methodology. They are executed either by a function or program designed in MATLAB.

The preliminary training in this chapter was to make course adjustments to the current model and as an indication that the current methods could be successful. During this initial training, the convergence of each network was analysed.

4.2 Preprocessing

Preprocessing of the training data is completed by a function and two programs. All computationally significant preprocessing is completed before the learning process. The preprocessing techniques for the test data are identical. However, it is achieved within the test programs to ensure new data is easily evaluated.

There is some preprocessing done in conjunction with ANN training. This processing is

in regards to creating matrix inputs and normalisation. As these are not a computational burden, there was no requirement to have them performed before the learning process. This section will explain the preprocessing cropping function, the dimension reduction algorithm, and the data augmentation algorithm.

4.2.1 Cropping Function

As discussed in Chapter 3, all images received are four refractive map images from a Pentacam. All data includes the axial/sagittal curvature (front), corneal thickness, elevation (front), and elevation (back) these four topographic maps are the only part of the image that is of interest for this research. Therefore, it is necessary to first isolate these four topographic maps from the rest of the image to ensure that irrelevant data is not presented to the ANN. All images follow an identical format and are initially 1200 by 840 pixels, discriminating unwanted information is also advantageous as it reduced the number of required inputs for the ANN.

This preprocessing step had to be automated and repeatable as the final model must be able to take an image not used for training and assess its probability of KC for that patient. A cropping function that is called by other programs was created to ensure repeatability and consistency. This implementation of this function is shown at Appendix G. The cropping function receives the refractive map patient image as an input. An input example is demonstrated by Figure 4.1.

The cropping function will process the input and provides an output of four cropped images. Examples of two outputs are shown by Figure 4.2. Each image is one of the topographic maps and all irrelevant information removed. The outputs are all the same size and aspect ratio for consistency in network inputs. This function is called during the training and testing phase of this project.

If this function were utilised in the future for topographic images from differing imaging equipment, modifications would be required.

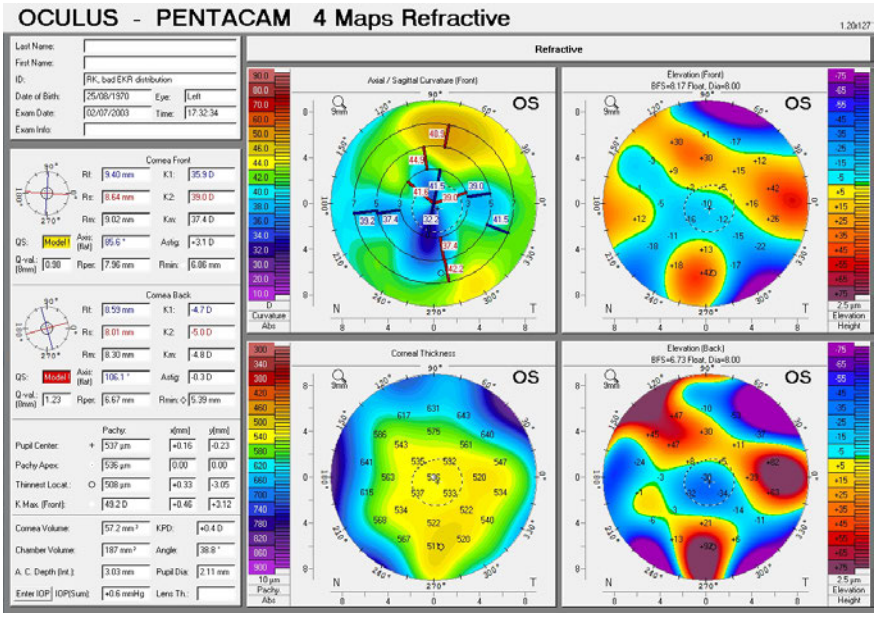


Figure 4.1: Control training image example. Initial image requires irrelevant information to be removed.

4.2.2 Dimension Reduction and Sorting

Most preprocessing is before the training process for efficiencies. The dimension reduction and sorting algorithm finds all patient images in the control and KC training folders and automatically processes them. This processing includes cropping, reducing channels, sorting, and saving all outputs. The MATLAB code for this algorithm is shown at Appendix H.

This program uses Pentacam images, as shown by Figure 4.1. The input image is passed to the cropping function. The outputs of this function are reduced to a two-dimensional image representing each of the RGB channels and greyscale. All of this processed data is sorted and saved into identifiable folders for access in other algorithms.

This process results in each input being converted to 16 different outputs and stored for future reference. This process will iterate through until all images in the data training folders are finalised. Extra data could easily be added for training at any stage throughout this project. The only requirement was to add or remove data from the training folders and execute this program.

(a) Corneal thickness.

(b) Elevation front.

Figure 4.2: Preprocessing output example with superfluous information removed. This shows 2 of the 4 outputs supplied by the cropping function.

4.2.3 Data Augmentation and Scaling

The algorithm described in Section 4.2.2 reduced all data from a three-dimensional RGB image to multiple two-dimensional images. This dimension reduction reduces the computational burden required for training. This Data Augmentation and Scaling algorithm further reduced that computational burden by scaling the training data. Once all the images are scaled, data augmentation is performed to ensure the available dataset is used efficiently. The MATLAB program to implement this algorithm is shown at Appendix I.

The scaling factor was varied throughout this research to determine the best compromise between information and the required processing time. An adequate compromise was reducing each image by a factor of ten. This reduction still provided enough information for the network to converge, and the processing time was manageable.

The data augmentation was achieved by rotating each image 90°. All data is then labelled at each rotation as either KC or control before being saved in the last training folder. This data augmentation is not just necessary to use data efficiently; it also helps prevent the network from identifying unwanted characteristics during training. It exposes the system to variations of the current images and is particularly applicable to this project as the region of interest is a circle.

4.3 Artificial Neural Network Architecture

This section will explain the algorithms and functions required for training and testing the final model. There are two programs described which were developed to train and integrate the ANNs for the final model. There is a program used to test the complete model, and finally, there is a function used by all to calculate the ANNs forward pass. MATLAB is used to implement all algorithms.

There is one additional program created that was used for testing the individual maps before creating the final model, however, as the programs described here were adapted from this model it is not included as it would be repetitive.

4.3.1 Forward Propagation Function

This forward propagation function processes one complete forward pass of the ANN. The function has been built for an ANN with three hidden layers and is shown by Appendix J. However, previous versions of this function had less, and if required, extra layers can be added relatively easily.

The function accepts the input as a single column matrix and requires the current weights and biases to calculate the forward pass. The total neurons per layer can be increased with no changes to this function. When changing the number of neurons, there must be careful consideration to ensure network architecture remains true. This consideration is to ensure matrix manipulation can be performed. These considerations are handled in the programs that call this function.

As was stated in Section 3.6.2, this project used the two-activation function sigmoid and swish. During preliminary training, the swish function would begin converging approximately ten epochs quicker in comparison to sigmoid. However, it did not show any significant improvements in total converging time. Hence, the added complexity of the swish function was deemed unnecessary for this specific dataset and network architecture.

The output of this function is the prediction of the network based on the input, weights, and biases. Furthermore, it can provide the results for all neurons in each layer. The output of each neuron was used during backpropagation to determine the cost.

4.3.2 Individual Topographic Map Training

The topographic map training algorithm was created to train four independent ANNs. Each ANN learns on a single RGB dimension for each topographic map. The implementation of this algorithm in MATLAB can be seen by Appendix K.

This program can train each ANN using either the red, green, or blue RGB channel. This choice can be modified by changing variables before execution. The first steps of this program are to perform some final preprocessing of all training data.

Each image is converted to a single column matrix and normalised. Each input is labelled by concatenating a trailing one or zero (KC or control). This label informs the network of desired output and labelling is based on the previous preprocessing naming convention. The data for each map was then combined to create four complete input matrices. For efficiency, all these steps were done before training.

The program trains each of the four ANNs, one at a time. All the steps are identical. The only difference is the input matrix and storage location of the final weights and biases. Hence, one network training process is explained here.

The program initialises and sets all parameters before the training commences. The weights are initialised using the Xavier Technique, and the biases are all made zero. An input is randomly selected, and the network gives a prediction by calling the forward propagation function. The desired output is compared to the guess and backpropagation is used to create small changes to the weights and biases of all layers.

Backpropagation was explained mathematically in Section 3.6. However, it will be briefly described here to relate it to this program. The error at each neuron is calculated backwards, beginning at the output layer. The error is dependent on the neurons current weight and the calculated error of the layer in front. The gradient and partial derivatives are calculated using these errors.

The gradient is the cost for the biases, and the partial derivative is the cost of the weights. Finally, the last step of backpropagation is updating weights and bias. The cost and learning rate is used for this updating.

The single forward and backwards pass completes one iteration. A different input is

randomly selected to repeat this process. This iterative process is completed until all training data has been processed, and this ends one epoch. The epoch is repeated until the network converges. The closing step of this program is to save weight and bias matrices for use with the final model.

4.3.3 Integration and Final Model Training

The integration and final model training algorithm was developed to integrate the topographic map networks and to train the final model. This integration uses the outputs of the four topographic systems and uses them as inputs for another ANN. This algorithm implementation is shown by Appendix L.

The fundamental ANN architecture and techniques are the same as the individual topographic map networks. The notable changes are the number of inputs and neurons. The training is by accessing the weights and biases for the four previous networks, calculating the forward pass for one complete preprocessed Pentacam image, and making adjustments based on desired results. Once this ANN converges the weights and biases are saved for the final model.

The forward and backward passes are all achieved as previously discussed. The result after the previous training and this training is five complete sets of weights and biases, which are integrated to create the final model.

4.3.4 Complete Model Test

This complete model tests algorithm is designed to take new data not used during the learning process and provide a prediction. The input is a Pentcam four maps refractive image, and the output is a prediction of the patient having KC or not. Additionally, it will give a model certainty of that prediction. This algorithm was implemented as a program in MATLAB and is shown by Appendix M.

The preprocessing techniques described in Section 4.2 and Section 4.3.2 is all included within this program. Once all training is complete this program, the weights and biases, the cropping function, and the forward propagation function are all that is required to assess new data.

This program will not be explained in detail as it uses the same methods as the previous algorithms. A notable feature is if the model predicts a certainty at or below 60%, it will not offer a prediction. An untrained model, on average, gives a certainty figure of 50%. Hence between 50% and 60% is not much better than a guess.

This model test program created the final results for this research. Additionally, it provides the final output and select preprocessing step outputs which are integrated with a mobile app which is described in Chapter 6.

Preliminary training and significant testing were performed to find the parameters, dimensions, and best techniques for this final model. This training and testing are in Section 4.4 and Chapter 5.

4.4 Preliminary Training

As was discussed in Section 3.5 to reduce processing time, the three-dimensional RGB data was reduced to two-dimensional images. Reducing the image to a single RGB channel achieves this reduction. Preliminary training was performed to ensure this was an appropriate method and to make course adjustment to ANN parameters.

The preliminary training was used to confirm the network will converge. It does not give a clear indication of system performance with new data. This section will establish if an ANN will converge on each of the four topographic maps independently and on which dimension. The algorithms used include all the preprocessing programs, the individual topographic map training program, the cropping function, and the forward propagation function.

4.4.1 Corneal Thickness

The corneal thickness map was used for the first preliminary training as visually it appeared that the green channel would differentiate between a patients image with KC and a patient image without KC. Figure 4.3 shows corneal thickness maps, one is of a patient with KC, and the other is a patient without KC. This figure highlights the dominance of the green channel.

The corneal thickness maps were used to train the network and determine the most suitable parameters. The images were made invariant to all RGB dimensions except green in the first instance. This selection would allow the ANN to converge the easiest, and then the learnt adjustments can be used to train systems on other RGB channels and topographic maps.

(a) Control image.

(b) Keratoconus image.

Figure 4.3: Corneal thickness maps.

The ANN comprised of three hidden layers. The number of neurons of each layer reduced by two-thirds of the previous layer. The preliminary training of the corneal thickness data began with a system of 841 inputs, 561 layer one neurons, 374 layer two neurons, 249 layer three neurons, and two outputs. The network training method was to alternate output activation based on the labelled input of KC or control.

In the first instances the learning rate was 0.1. This rate was significantly too high as the MSE per epoch would oscillate, and the network would not converge. Equation 3.5 defines the MSE in the context of this research. As the system would backpropagate on every forward pass reducing this overshooting could be achieved by significantly lowering the learning rate or changing the backpropagation technique. First, the learning rate was reduced.

In most cases, the ANN would converge with a new learning rate of 0.005. Although, at times the system did appear to get stuck at a local solution or continually hunt for the optimum solution. Figure 4.4 shows an example of when the system could not find the optimum solution.

The networks complexity was increased to ensure it would converge more regularly. The first hidden layer neurons were 1682, and each preceding layer was two thirds less. This increase in neurons increased convergence time. However, it no longer found local solutions

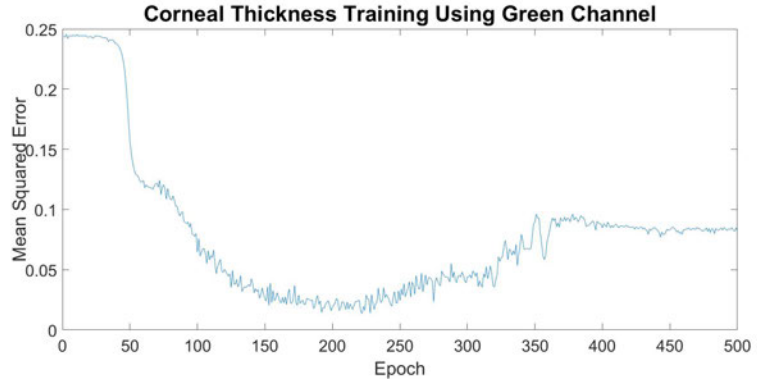


Figure 4.4: Example of an unsuccessful corneal thickness training attempt with green channel.

or oscillated. Independent training was done 50 times, and the added complexity network converged on all occasions. Figure 4.5 shows a sample of this training.

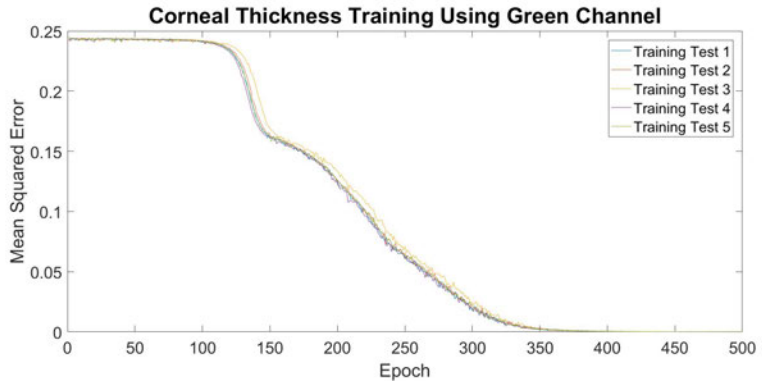


Figure 4.5: Corneal thickness training with green channel. Sample of 5 individual training attempts, showing convergences on all.

As was discussed, the backpropagation mode is another way to reduce oscillations or prevent local solutions. However, with the added complexity of the network, the batch technique would not converge within 500 epochs, and the stochastic mode would converge but did not show adequate improvements. As Figure 4.5 showed convergence on all occasions for this specific data, the online backpropagation mode is acceptable.

As the current network model will converge, it was necessary to determine if the green channel as initially thought is the best channel for training the corneal thickness data. To learn how each RGB channel will respond. The ANNs parameters were held constant, with only the input dimension changing.

All RGB channels were trained separately with a greyscale input for comparison. Figure 4.6 shows an example of this training.

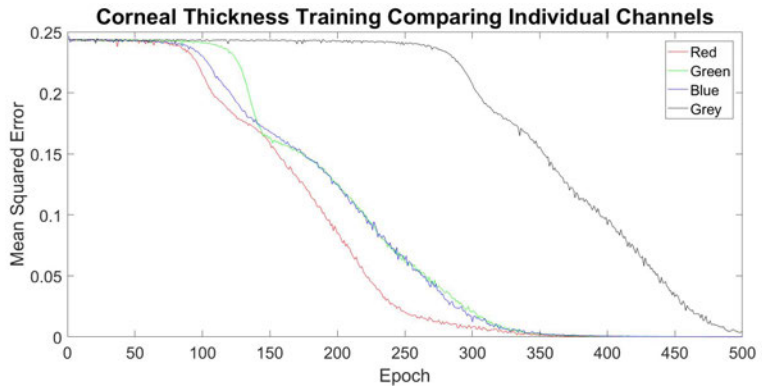


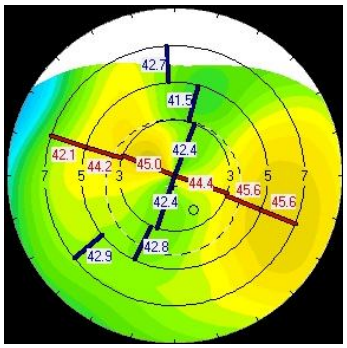
Figure 4.6: Corneal thickness ANN training comparing RGB channels and greyscale. Four training comparisons performed on corneal thickness dataset. The data is discriminated to only include a red, green, blue, or greyscale channel.

Figure 4.6 shows that the three RGB channels will all converge between approximately 350 to 400 epochs. In comparison, the greyscale image had still not wholly converged after 500 epochs. This convergence is promising, but it is not an indication of performance on unique data.

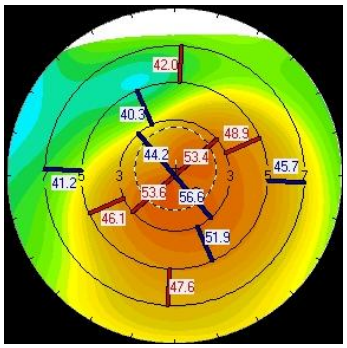
To further determine which dimensions this network can be made invariant to and perform accurately, further testing is required. The results of further testing are in Chapter 5.

4.4.2 Axial / Sagittal Curvature

The sagittal curvature preliminary training used identical network parameters as the corneal thickness training. Figure 4.7 shows an example of sagittal curvature images showing a patient with KC and a patient without KC.



(a) Control image.



(b) Keratoconus image.

Figure 4.7: Axial / sagittal curvature maps.

Figure 4.8 shows the results of preliminary training for sagittal curvature data. It showed the red and the green channel converged between 300 and 350 epochs. However, the blue channel did not converge after 500 epochs. Interestingly the greyscale images did converge with this data, but it was less efficient than the red and green channels.

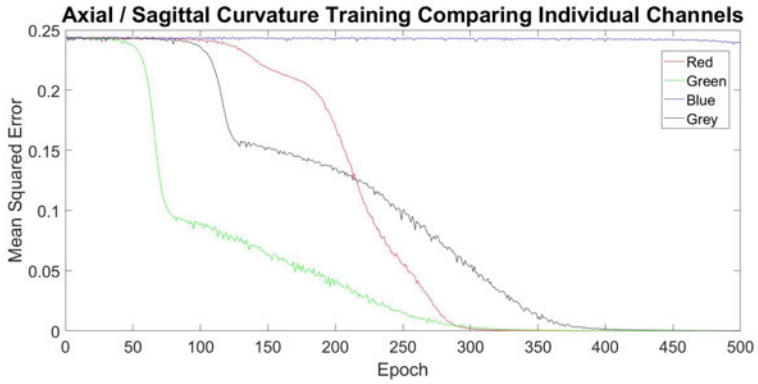
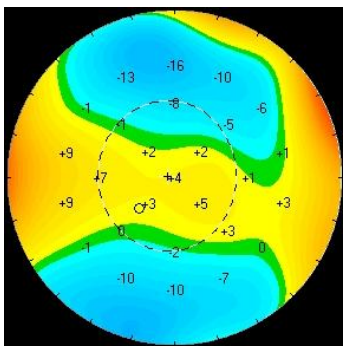


Figure 4.8: Axial / sagittal curvature ANN training comparing RGB channels and greyscale. The data was discriminated to only include the red, green, blue, or greyscale channels.

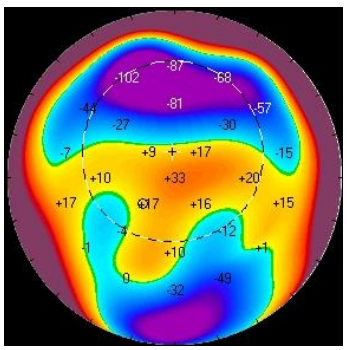
As per the corneal thickness preliminary training, it is not possible to determine from convergences which channel will perform best on data not used for training. Further testing on sagittal curvature images is in Chapter 5.

4.4.3 Elevation Front

The elevation front preliminary training used the same network parameters as the previous training. The elevation front images of patients with KC are visually distinct from the last topographic maps. Figure 4.9 shows an example of inputs before dimension reduction used for this training.



(a) Control image.



(b) Keratoconus image.

Figure 4.9: Elevation (front) maps.

Figure 4.9 shows there is significant differentiation between patient images with KC and patient control images. This differentiation is synonymous for a large portion of the elevation front training data. However, there is still a small portion where KC and control patient images look similar.

Figure 4.10 shows the training performed on the elevation (front) topographic map inputs. Unsurprisingly the front elevation topographic maps on the blue channel show quick convergence when compared to the training of previous data. This visual difference of blues in Figure 4.9 has translated to quicker network learning.

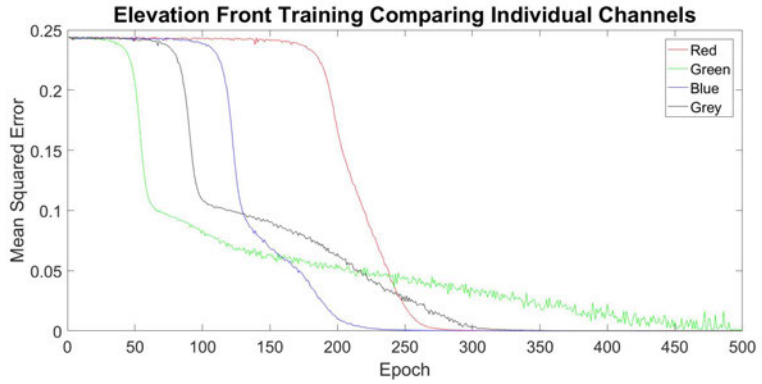


Figure 4.10: Elevation (front) ANN training comparing RGB channels and greyscale. The data is discriminated to only include the red, green, blue, or greyscale channel.

In Figure 4.10 it can also be seen that the green channel did not converge efficiently. This inefficient convergence is surprising as the images did have visual differences of greens. This training was repeated multiple times to confirm the observation. All additional training showed the green channel not entirely converging before 500 epochs.

It seems that the blue channel would be the optimum solution for this map. However, further analyse was still performed with unique data to the training. This ensures the right dimension invariance for the final model. This testing is described in Chapter 5.

4.4.4 Elevation Back

The elevation back topographic maps are similar to the elevation front when comparing images of patients with KC to images of patients without KC. Figure 4.11 shows typical KC and control topographic maps from the elevation back training data.

A visual analyse of the elevation back training data shows a higher percentage of these

(a) Control image.

(b) Keratoconus image.

Figure 4.11: Elevation (back) maps.

images will be harder to differentiate than the elevation front data. Figure 4.12 shows the training on the three channels and greyscale.

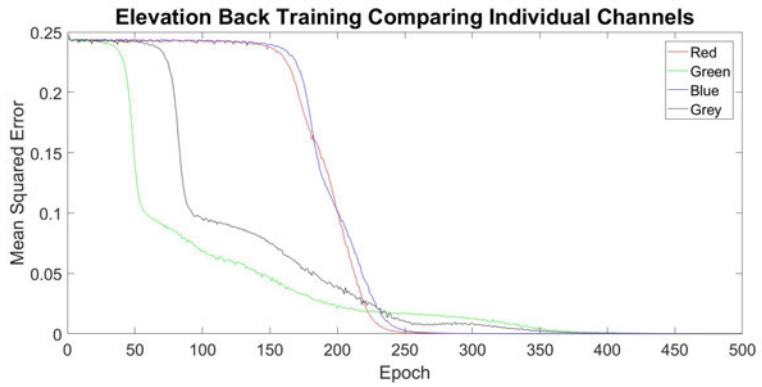


Figure 4.12: Elevation (back) ANN training comparing RGB channels and greyscale. The data is discriminated to only include the red, green, blue, or greyscale channel.

Predictably the elevation back data as a whole converged slower than the elevation front. Although the red and blue channel still converged within 300 epochs. The green and greyscale also showed convergence. As with all of the previous topographic maps, this convergence does not give a clear indication of performance on new data, and more testing is required.

4.5 Chapter Summary

This chapter has shown all the relevant programs and functions for project implementation. The complete model test program does repeat a portion of the preprocessing programs. However, it is designed that way so that this program, the trained classifiers,

and two functions are the only requirements to make predictions on new data.

The preliminary training showed encouraging results and confirmed that the current methods were appropriate. This training highlighted the required adjustments to the networks architecture and techniques. Further testing and analyse is necessary to ensure the most applicable inputs are selected for all ANNs.

Chapter 5

Training, Validation and Analysis

5.1 Chapter Overview

This chapter describes the training and validation performed on all the ANNs employed in this research. The results of this training and validation were used to determine optimum parameters for the final model. These results are presented for each topographic map. Furthermore, this chapter describes how the available data was utilised efficiently.

This chapter uses the results of the final model to adjust parameters and techniques to achieve a solution. The solution is analysed to show that the results provide a proof of concept for the project's hypothesis and highlights the limitations.

5.2 Individual Topographic Training and Validation

As was discussed in the literature review in Section 2.7, typically, a validation dataset would be used as an unbiased indication of model fit. This validation dataset is primarily used to indicate when overfitting of the model has occurred during training, while the lack of data prevented a standalone validation set for this project, a similar approach was employed.

For efficient use of data, this research used a technique called the leave-one-out cross-validation method (Kohavi et al. 1995). Precisely for this project, it was accomplished

by training the network with one image isolated and testing the system at specific epoch points with this isolated image. This method is costly to compute. However, it was necessary as it makes regular predictions, and as the number of epochs increased if the data began overfitting, there would be an indication within these results.

The average certainty of the networks forward pass is used for results in this section. Every forward pass the model gives a confidence figure between 0% and 100% for KC and control. The mean of these values for the correct answer was the average certainty.

This assessment gave a prediction every two epochs and its current certainty. Testing was repeated for all the data until average confidence from 0 to 450 epochs was shown. The examination was for each RGB channel and each of the four topographic map networks. As Chapter 4 showed the greyscale on each of the topographic parameters would either result in the system not converging or converging less efficiently, testing was only on the RGB dimensions.

The testing program used the same processes as the algorithms discussed in Chapter 4. The only addition to the procedures discussed was extra loops to automate the process and additional data capture.

The results for each topographic map networks are shown in this section. These results are dependent on the initial data but provide reliable statistics to make an informed choice on final model parameters.

5.2.1 Corneal Thickness

Figure 5.1 shows the results of average certainty and validation for corneal thickness on each RGB channel.

This result shows that the green channel is superior to the red and green for the corneal thickness network. Following the test, it gave an average certainty of close to 80%, and overfitting was not evident. Interestingly, when comparing these results to Figure 4.5, the green channel has approximately a 15% higher average certainty to the blue when convergence occurred at similar times.

The corneal thickness ANN in the final model was made invariant to the red and blue

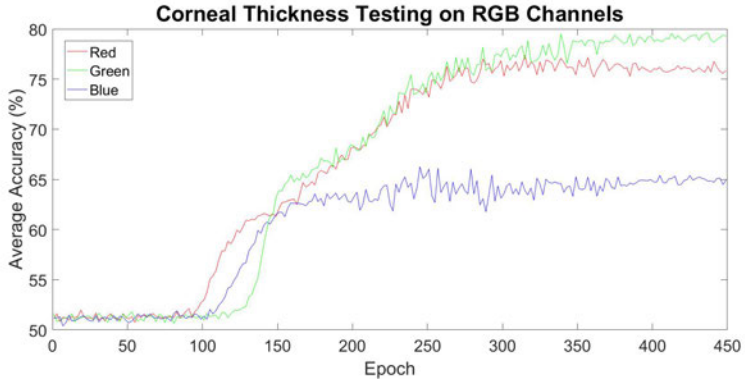


Figure 5.1: Corneal thickness testing comparing RGB channels and average certainty per epoch.

channel.

5.2.2 Axial / Sagittal Curvature

Figure 5.2 shows the results of validation for the sagittal curvature ANN on each RGB channel.

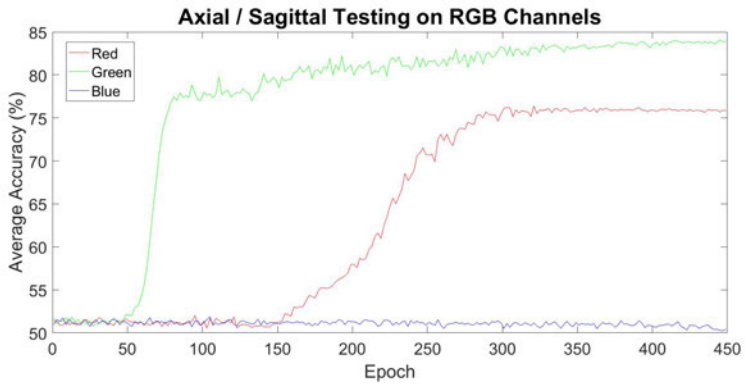


Figure 5.2: Axial / sagittal curvature testing comparing RGB channels and average certainty per epoch.

The sagittal curvature network gave a correct average certainty of close to 85% following this test and no evidence of overfitting to the training data. Comparing these results to the networks training convergence in Figure 4.8 it is not surprising the blue channel did not perform better than a guess. Additionally, the red channel results plateau and settle at 8% less average certainty than the green channel. This trend was impossible to predict solely based on network convergence. This result and the corneal thickness results underline the importance of testing and validation on unique data.

The sagittal curvature final models' ANN was made invariant to the red and blue channel.

5.2.3 Elevation Front

Figure 5.3 shows the average certainty results and validation for elevation front network on each RGB channel.

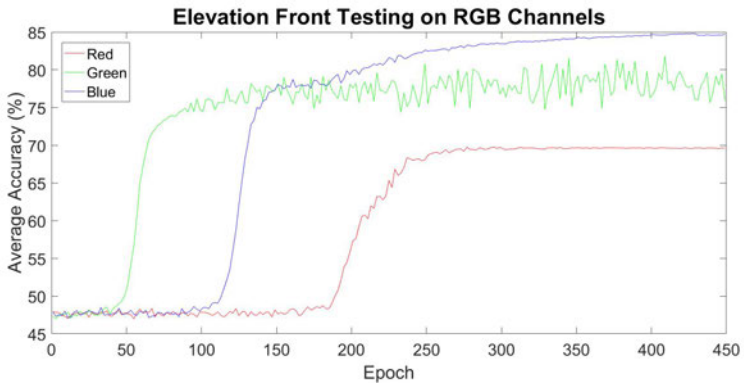


Figure 5.3: Elevation front testing comparing RGB channels and average certainty per epoch.

This testing shows the blue dimension on the elevation front network gives approximately an 85% correct average certainty. Additionally, the blue channel does not indicate any overfitting during this training. Interestingly, the red channel significantly underperformed, which was not evident by observing the MSE during preliminary training in Figure 4.10.

The elevation front ANN was made invariant to the red and greed channel for the final model.

5.2.4 Elevation Back

Figure 5.4 shows the average certainty results and validation for elevation back network on each RGB channel.

The green dimension outperformed the others in this testing. Furthermore, it was still increasing in average certainty after the test. Hence, in the final model, the elevation back network will be trained over more than 450 epochs.

The elevation back network was made invariant to the red and blue channels for the final

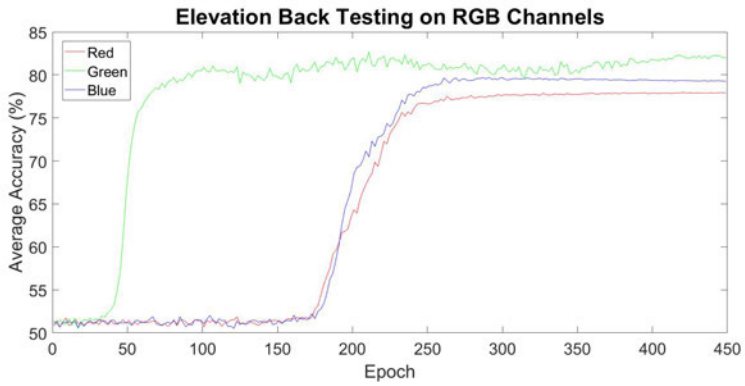


Figure 5.4: Elevation back testing comparing RGB channels and average certainty per epoch.

model.

5.3 Final Model Results and Analysis

The previous section did not only validate the training dataset but also gave critical results to define and create the final model. The results were used to determine which RGB dimension is used during training the four topographic ANNs. Utilising a single optimum dimension for each map ensures irrelevant information is not presented to the ANN, similar to what was seen by the blue channel in Figure 5.2, and computational effort is reasonable. The four ANNs were prepared using the algorithm shown by Appendix K.

After the final model training, the four ANNs were integrated using the algorithm shown by Appendix L. Finally, this final model was verified on unique data not used during the learning process. Appendix M was used to complete the final model testing.

As was discussed in Chapter 3 and 4, the final model is trained to differentiate between patients with KC and patients with other corneal disease or refractive conditions. For these results, other patient images are referred to as control. Furthermore, if the model certainty for an image is at or below 60% it will not make a prediction. The results of the two training sets are discussed in this section.

5.3.1 Initial Training Set

The trained final model was verified on data not used for the learning process. Firstly, the model was shown 49 images of patients with KC. The model correctly identified 87.8% as KC. The images that were not correct predictions where all below the certainty threshold. As such, the network did not give any false predictions. The models correct predictions were with a certainty range of 71.7% to 98.1%.

Unfortunately, of the 45 unique control images tested, the model did not yield fruitful results. The model gave approximately 60% correct identification and produced several false positives of KC. Due to these unwanted results further testing and analysis was performed.

A theory was that using data augmentation to increase the inputs to the final model could improve results. All test control images were rotated at 90° increments and processed through the current model. The results are in by Table 5.1.

Table 5.1: Initial training results for control data. Results are shown as a percentage of correct identification at varying input angles.

Data Augmentation Testing on Control Data			
Rotation (degrees)	Keratoconus (%)	Uncertain (%)	Control (%)
0	20	20	60
90	20	22.3	57.7
180	22.3	20	57.7
270	17.7	24.6	57.7

The same techniques were used for the KC test data. However, all results remained identical at each rotational angle. Table 5.1 shows that at best, this model will give 60% correct identification of a control image. These results highlight that the final model has not identified unwanted characteristics and is mostly invariant to rotation. The control image results were not great at this point, but the invariance to rotation is very promising and shows data augmentation at the learning process was successful.

Data augmentation does not present any significant improvements when used in this context and at this stage. This technique which would make the final model more complex

is not beneficial. Hence, other methods would need to be employed to improve the results.

During the integration training, it was observed that convergence occurred within 100 epochs. Therefore, overfitting was suspected. Even though Section 5.2 showed each map individually trained adequately, it appeared when combined, overfitting occurred. A random sample of the KC and control test data was switched to the training set to determine if this would alleviate the overfitting phenomenon. This reduced the test data and confidence of results. However, it will show the benefits of increasing available data when training an ANN.

The results shown in Section 5.3.2 will highlight that extra training data minimised overfitting.

5.3.2 Final Training Set

Test data was removed for training leaving 44 images for testing. However, this increased the training data to a total of 134 images, and with data augmentation, on each cropped topographic map, the model was trained on 2144 images. The final model learned with these 2144 images and tested again.

With this additional training data, the model correctly differentiated between control and KC 90.9% of the time on the 44 test images. The training and testing were repeated several times, and these results were repeatable for this specific training and test datasets.

The images of patients that were not predicted correctly were either control images incorrectly identified as KC or the model was not above the certainty threshold. All KC images were correctly identified.

5.3.3 Analysis

The result of 90.9% is a suitable proof of concept that an ANN can be used to differentiate between patients with KC and patients without KC. Furthermore, as the control images used were of patients with other corneal conditions or refractive issues, it is not surprising that some were incorrectly identified as KC. Figure 5.5 shows a control image that was incorrectly predicted to be KC.

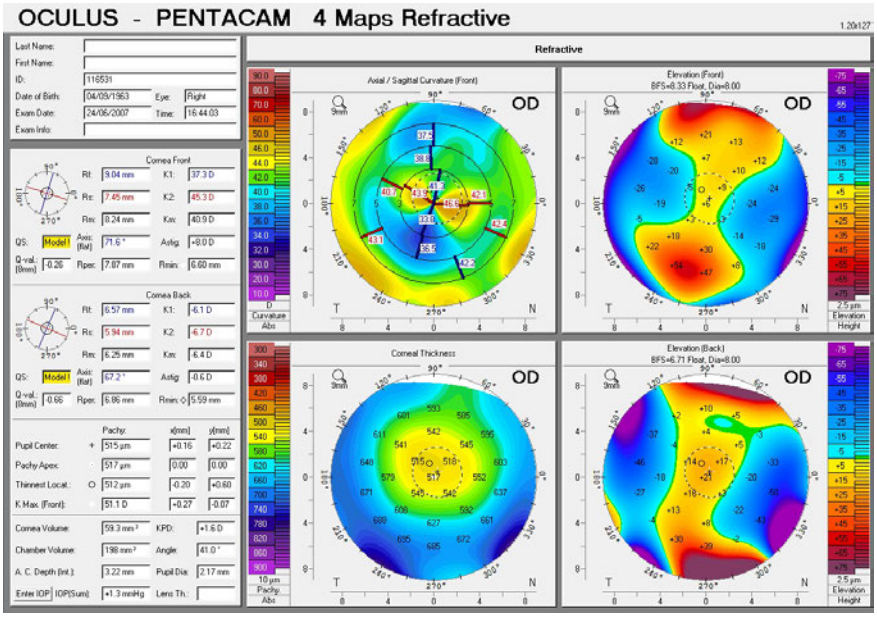


Figure 5.5: Control image incorrectly identified as Keratoconus.

This actual patient diagnosis of this image is unknown. Although, when observing the elevation front and elevation back topographic maps, these are synonymous to some of the patient images with KC. Furthermore, the corneal thickness map shows a similar topographic shape to that of patient images diagnosed as mild KC. These comparisons are with the patient images within the training data. With the complete network learning from this data, it is impressive the model has performed as well as it has. As the model has shown, it can differentiate between patients with KC and different conditions, subject to data limitations, it would be reasonable to assume similar techniques could be expanded for other corneal abnormalities.

There are indications that a small correlation between the severity of KC and the model's certainty is evident. This correlation was observed when severity versus certainty results were plotted. However, further data is required to confirm or dismiss this hypothesis.

If this model used significantly more data and with the addition of healthy cornea images in the learning phase, the performance could be improved. The results in this section have shown a clear benefit of the acquisition of larger training sets.

5.4 Chapter Summary

This chapter has shown that after variations in the techniques used to create the final model, that it can differentiate between a patient image with KC, and a patient image without KC 90.9% of the time. The result show adequate proof that an ANN can be employed to detect KC and could be expanded to detect other corneal abnormalities with suitable data.

These results are promising, indicative that the research hypothesis may be sustainable. Subject to data limitations, further work would show improvements to this research.

Chapter 6

Android Application

6.1 Chapter Overview

To help prove the portion of the hypothesis that a mobile device an be used to diagnose corneal abnormalities, a mobile app was created. This application is under development. However, in its current state, it presents some proof, and it shows select stages of processing.

This chapter will explain the outputs that the mobile app prsents to the user and a broad overview of its development.

6.2 Description

The development of the mobile application was done in Android Studio. The mobile app is currently compatible with any Android device with an operating system at or above version 7.0.0 (Nougat). For this research, the app was installed and tested on a Samsung Galaxy S7 and a Samsung Galaxy Tab E.

The app shows some of the preprocessing steps to the user and then provides the prediction, certainty, and actual ophthalmologist diagnosis on the final screen. It allows the user to select an uploaded Pentacam image. The app will then step through three screens showing the user the outputs after the first preprocessing step, then after dimensionality

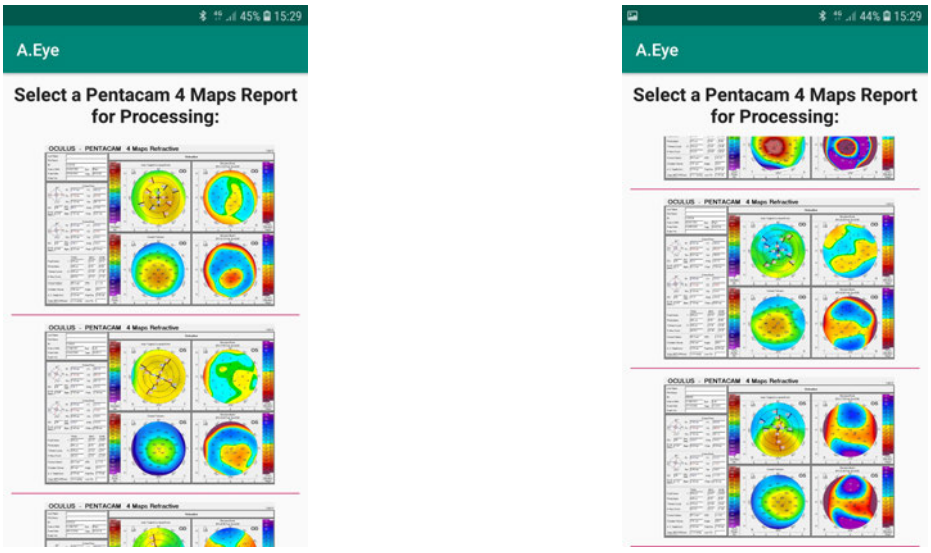
reduction, and finally, the results and initial user selection.

The images uploaded to the device was data not used for training the classifier. For this research, Pentacam image were placed internally within the app. Having the data stored within the app prevented the need for network or internet access. However, in the future, the app should be designed to access an online database. Having it connected to an online database will make the app more flexible and user-friendly.

6.3 User Interface

This section will show how the user interfaces with the mobile app. Screenshots from a Samsung Galaxy S7 were taken to aid in the apps explanation.

The user interfaces with the mobile app by selecting one of the uploaded Pentacam images with the devices touch screen. Figure 6.1 shows the main screen when the app is executed.

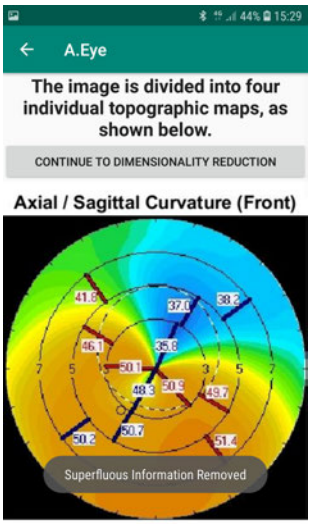


(a) Top of display.

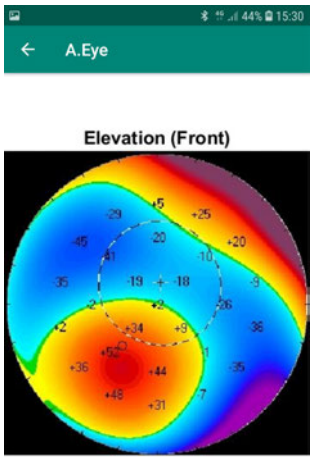
(b) After user has scrolled.

Figure 6.1: Mobile app main screen. User can select Pentacam image for processing.

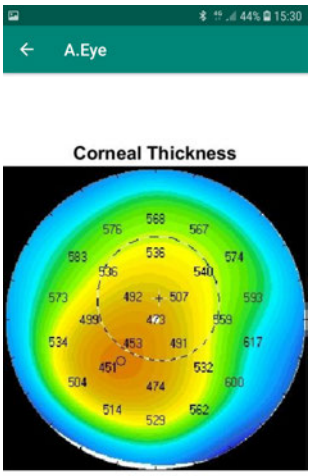
The user can scroll through all Pentacam images that are internal to the app. The data is not labelled as KC or control as at this stage of processing the diagnosis would be unknown. Once the user clicks on their desired image for processing the next display will be shown. Figure 6.2 shows the preprocessing cropping function output of the selected image.



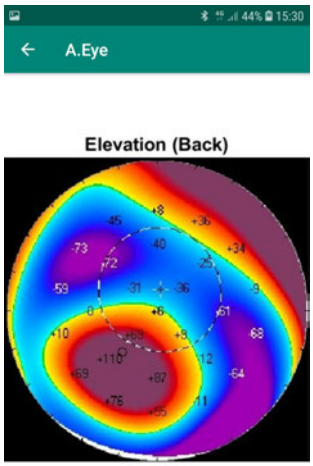
(a) Top of display.



(b) After user has scrolled.



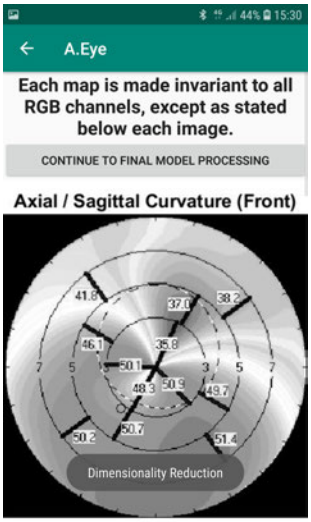
(c) After user has scrolled.



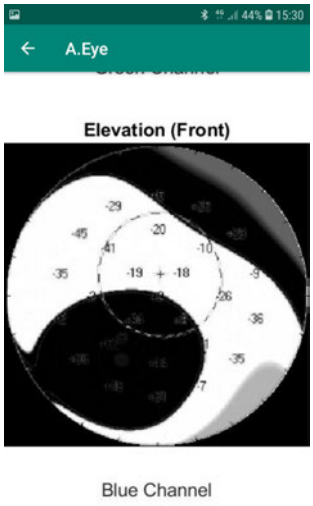
(d) After user has scrolled.

Figure 6.2: Mobile app first user display. Output after using cropping function on selected image.

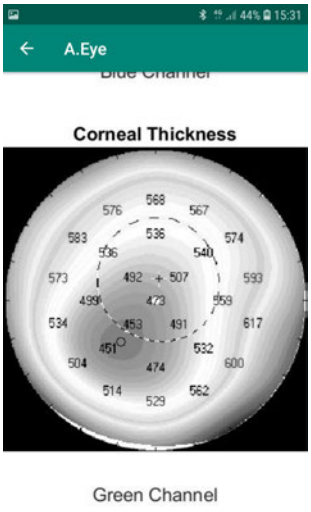
The user can scroll down and view each of the topographic maps without the redundant information. To advance to the next processing step, the user selects the continue button shown by Figure 6.2a. The app will then display the output after further preprocessing shown by Figure 6.3.



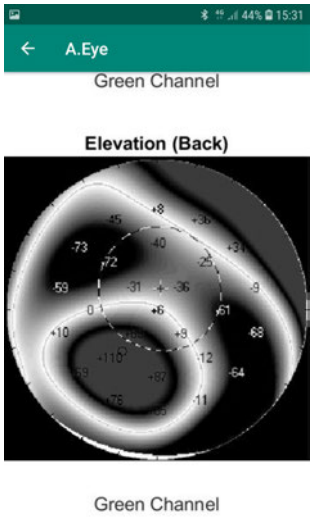
(a) Top of display.



(b) After user has scrolled.



(c) After user has scrolled.



(d) After user has scrolled.

Figure 6.3: Mobile app second user display. Output after data reduced to single dimension.

The app shows the result after the data is reduced to a single dimension. Furthermore, it highlights which aspects the network will be made invariant to for each map. The user can scroll to view all data outputs and can continue to final model processing by selecting continue as shown by Figure 6.3a. Figure 6.4 shows the final model processing for the selected image.

Figure 6.4: Mobile app model prediction.

Figure 6.4 shows to the user the models prediction, the models certainty, the actual patient diagnosis, and the original image selected. This app shows how a mobile device can be utilised for similar processing. In further development displaying the processing steps may not be necessary, and the actual diagnosis would be unknown for new patients. However, this shows what an app like this in the future could be modelled off.

6.4 Backend and Tools

This section will give a broad overview of the tools needed to create the app and the structure of the backend. For further information, the author should be contacted to request access to the GitHub where it is hosted.

The integrated development environments (IDE) to make this mobile app included Android Studio and Repl.it. Android Studio was the primary IDE where the mobile app

project was developed, and Repl.it was used to transform some JavaScript Object Notation (JSON) structures.

The languages used for the app included Kotlin, JavaScript, and XML. Kotlin was for the backend, XML was for the styling of each display, and JavaScript was used in conjunction with Repl.it to transform the JSON structures.

The testing at various stages of production was with the Android Studio Emulator. The mobile devices used with the emulator were a Google Pixel 3 and a 7 inch Android Tablet (generic). The mobile app has been tested on software versions from 7.0.0 to 9.0.0. Hence, correct operation is not guaranteed outside of this software scope. Testing with the emulator was conducted before uploading the app to any hardware devices.

The backend includes the database, source code for each page, and layout structures for all displays. The image and database information is included as a JSON string, and the getters and setters of the images and content are stored dynamically. Dynamic allocation of the database reduced the error with hardcoding each image and allows the database to be modified easily. As the loaded data is increased, the user list is automatically updated. Furthermore, this will make access to an online database easier to implement.

The source code for each page is in Kotlin, and it will read off its corresponding XML layout structure to populate the images to display on that page. The first page is described here, and the code can be found in Appendix N. However, it is similar to the other source code, so the description is mostly transferrable to all.

This list of images for the main page is via the XML structure and access to the database. When the user selects the picture, it is passed using the onItemClick function via the putExtra. The imageModel data is then passed to the next pages source code for processing. The model to have images for each concluding display is also allocated within this first source code, including the text for the last page.

6.5 Chapter Summary

This chapter has highlighted how a mobile app could be used to diagnose corneal abnormalities. The app created for this project allows a user to select a Pentacam image and

view the prediction. Additionally, it will show some select preprocessing steps that the model will perform.

The tools and structure of the app were shown and a sample of the source code listed. Showing all source code and structuring is not feasible for the mobile app project, but access can be granted on request.

Chapter 7

Conclusion and Further Work

7.1 Chapter Overview

This chapter will highlight the achievements of the project, give a conclusion, and make recommendations for further work. The conclusion highlights the fact that the original hypothesis is sustainable and that with additional work diagnosing corneal abnormalities may be more straightforward in the future. The project objectives will be evaluated individually to define what was achieved. The section on further work gives recommendations to gain greater confidence in this research results and initiatives to expand on this work for the future.

7.2 Conclusion

This research project has shown the hypothesis that a mobile device can be used to diagnose corneal abnormalities is feasible. This diagnosis would be without the need for specialised imaging equipment or review from an experienced ophthalmologist.

This project approach was differentiating between patients with KC and patients without KC as proof of the concept. Patient data from ASK's Pentacam imaging equipment was used in conjunction with preprocessing techniques and machine learning algorithms. The final model gave promising results and could differentiate between patients with and without KC 90.9% of the time using unique data.

These results are especially impressive as the data included many patients with other corneal diseases or refractive issues than KC. The fact that the model differentiated between patients with varying corneal abnormalities and not just between KC and a healthy cornea holds further weight to prove the original hypothesis.

Additionally, this research supported the literature in the theory of more extensive databases improving results and reducing overfitting for ANNs. When the training database for the project was increased, the model's correct predictions increased. This supports recommendations to increase the database in further work.

This research project also developed a mobile app. The apps intent was to offer further proof for the aim of the project and give an example of how a user may interface with a diagnosis app. The app showed the user select processing steps and gave the prediction based on the models' trained classifier. The developed app provides an adequate proof of concept and an excellent foundation that could be expanded on in the future.

Further work is required to use a mobile device as a standalone system and to definitively determine the methods that could be used for other corneal conditions. However, pending data availability, there are indications this would be successful.

With an extension of this research, it could be seen that diagnosing many corneal abnormalities could be achieved with less time and resources in the future. Early detection of many corneal conditions is paramount for early intervention to preserve sight and prevent corneal transplantation. These benefits could be especially noticed in the developing world where access to expensive imaging equipment and experienced ophthalmologists are rare.

7.3 Achievement of Project Objectives

The projects aim was to determine if a mobile device, in conjunction with ANNs, could be used to diagnose corneal abnormalities. To reach this aim, nine research actions were defined in Section 3.2.1. These objectives will each be evaluated for their success.

(i) Obtain patient images of corneas with and without KC.

As discussed in Section 3.4, this project utilised patient images from a Pentacam

provided by ASK. There were some delays and limitations to the data. This was due to ethical approval processes and arduous download procedures from the Pentacam. However, the data received was adequate to complete this research.

(ii) Adapt different ANNs for the project.

The generic approach to machine learning is described in Chapter 2 and Section 3.6, the specific algorithms suitable for the type of problem analysed here are detailed in Section 4.3. As was shown, ANNs of varying complexities were designed and implemented using MATLAB. The networks developed included hidden layers ranging from one to three with adaptable input and hidden layer neurons. These different networks used either swish or sigmoid activation functions, and the backpropagation techniques were online, batch, or stochastic modes. Having multiple models enabled the best parameters and methods to be used for the final solution.

(iii) Create algorithms for standardising and preprocessing of all images.

The approach to preprocessing is shown in Chapter 2 and Section 3.5, the specific algorithms suitable for the data in this research are in detail in Section 4.2. The implemented algorithms for preprocessing, sorting, and standardising were successful. The algorithms enabled a dynamic dataset. Training and test data could be added or removed, and the algorithms automatically sorted and processed it. Furthermore, they are modifiable to adjust preprocessing parameters or desirable inputs for ANNs.

(iv) Test compatibility of the ANN and preprocessing algorithms.

The testing of compatibility of ANN and preprocessing algorithms was performed concurrently with their development. This was shown in Section 4.4 by observing the models convergence. The examination was successful, and a significant outcome was that the input neurons and image data inputs could be easily modified for flexibility during data testing and verification.

(v) Determine the best network, the ideal parameters, and best training data.

The adjustments to the model's parameters are shown in Section 4.4.1 and training data is analysed in Section 5.3.1. The project chose the ANN techniques, settings, and training data based on indications of desired outcomes for the computational

effort. Data augmentation for the training data was used to reduce the likelihood of overfitting to limited data availability. After testing and validation of different ANN models, the final model used an ANN with three hidden layers, sigmoid activation function, and online backpropagation mode. Determining the best parameters and techniques were successful for this specific database and project.

(vi) Carry out extensive testing and determine networks performance.

This testing in this objective used the leave-one-out cross-validation method, and it was to determine the optimum preprocessing for each topographic map. The observations from this method are shown in Section 5.2. Each of the four topographic maps was assessed independently, and the RGB dimension that performed the best on unique data was used for the final model. This objective was very computationally exhaustive but directly contributed to successful final results.

(vii) Test independent data on trained ANNs.

The final model testing and analysis is shown by Section 5.3. The model was tested on unique data not utilised for the learning process. The final results were promising and showed that the model could differentiate between a patient with and without KC 90.9% of the time. Furthermore, there was an indication of a correlation between model certainty and KC severity. However, verification with more data is required to prove this hypothesis. Confirmation of the results has been recommended in further work.

(viii) Investigate reconstructing corneal topographic images from a mobile device.

Section 2.11 showed that the literature review highlighted there are current corneal topography mapping techniques for a mobile device. The technology currently available requires an attachment for the camera of a mobile, and in some cases, images are sent to a cloud-based server to construct the topography. The techniques show promise as an extension to the achievements in this research and further work has been recommended.

(ix) Create Android mobile application for demonstration.

The project successfully created an Android mobile app, as shown by Chapter 6.

The app allowed the user to select a Pentacam image, stepped through processing steps, and gave a prediction of KC or control. The app helped support the original hypothesis. However, further work is required for the app and recommendations have been given.

The achievements of the project, as outlined in these objectives, has covered all standard aspects of the project specification programme shown in Appendix A. Furthermore, the research has investigated the time permits step of determining suitability for a mobile device. Due to data limitations, the project could not apply the same research techniques for Fuch's Dystrophy, as stated in the time permits programme. There is scope for improvements in some objectives, and these are discussed in Section 7.4.

7.4 Further Work

This section will give recommendations for further work. The first suggestion includes expanding the current patient images database to verify the results of this project. Another recommendation is using a database of images from differing imaging equipment and exploring additional preprocessing techniques. Lastly, the report suggests integrating the model with mobile topographic technology and improving the app.

7.4.1 Data Acquisition and Verification

To better understand the implications of these results, future studies could address diagnosing other corneal abnormalities and improving the classifiers for diagnosing KC. As Fuch's Dystrophy is the leading cause of corneal transplants (Gain et al. 2016), this condition would be an ideal candidate for future studies.

Increasing the database of patient images would be pivotal to this further work, as this was the most significant limitation of this research. With more data, the confidence in results would be increased, and it could be definitively determined if there was a correlation between patient severity and the models' prediction certainty.

If more data from a Pentacam is made available in the future, the algorithms developed in this research can be directly applied. The algorithms were designed to process and

sort all available data autonomously. This allows future work in KC detection or other corneal abnormalities that follow the same methodology to be effortlessly achieved.

7.4.2 Data Diversification

CXL slows the progression of KC and can reduce the need for corneal transplants. This surgical procedure requires maximal corneal thickness to protect the corneal layers and is only suitable for patients in early stages of KC (Kymes et al. 2004). Early detection is, therefore, a prerequisite for this treatment.

As there are only two organisations in Australia with access to a Pentacam, a robust classifier that can diagnose KC early on less advanced equipment would benefit the maximum amount of patients (Gain et al. 2016). Therefore, additional work could be committed to supplementing the available database with corneal topographic maps from varying medical equipment. This is in contrast to this research using images solely from a Pentcam. This extra data would decrease the likely hood of the model fitting to unwanted characteristics that may be evident for differing technologies, and the model could be useful for diagnosing KC from images using a diverse range of equipment. However, there will be challenges with using different imaging equipment.

Data preparation will be vital in using patient images from different equipment. As such, the preprocessing algorithms will need to be flexible. The preprocessing algorithms in this research would be useful, but they will likely need to be supplemented with additional techniques. Furthermore, methods like pre-trained segmentation networks that were discussed in Section 2.7 may be helpful in this extra work to avoid repetitively training classifiers from the ground up every time new data is obtained.

7.4.3 Model and Mobile Integration

If training and verification is successful with a model that has learned off images from varying equipment, this could be combined with current mobile topographic technology. This topographic technology was highlighted in Section 2.11, and further work is required to determine if this would be suitable for automatic medical diagnosis.

Further work with the mobile app could be done to achieve additional flexibility and

efficiency. The app could be improved so it can access an image online as an alternative to it been uploaded to the device for processing. Additionally, it may be worth investigating the processing of images in the cloud. This would alleviate any limitations of processing power a mobile device may have, allow simple adjustments to the model design and easier integration with current mobile topographic technology.

As was discussed in Section 1.2.2, a high demand for corneas has resulted in a worldwide shortage of corneal donor tissue, and approximately 12.7 million people are on the corneal transplantation waiting list (Gain et al. 2016). Early diagnosis of KC and other conditions can reduce the need for transplantation. Mobile technology similar to what is proposed here may help with this unfortunate concern by diagnosing additional patients without access to advanced equipment.

7.5 Chapter Summary

It has been shown that the research objectives have been met and the research hypothesis, subject to data limitations, should hold with further work.

Further work is required and has been proposed. However, this project was successful within the scope and limitations of the research

References

Accardo, P. & Pensiero, S. (2002), 'Neural network-based system for early keratoconus detection from corneal topography', *Journal of Biomedical Informatics* **35**(3), 151 – 159.

Bishop, C. M. et al. (1995), *Neural networks for pattern recognition*, Oxford university press.

Bourges, J.-L. (2017), 'Corneal dystrophies', *Journal Français d'Ophtalmologie* **40**(6), e177 – e192.

Catania, L. J. & Nicolitz, E. (2018), 'Artificial intelligence and its applications in vision and eye care', *Advances in Ophthalmology and Optometry* **3**(1), 21–38.

Chellappa, R. (2016), 'The changing fortunes of pattern recognition and computer vision', *Image and Vision Computing* **55**, 3–5.

Committees, R. (2014), '"human research ethics committee", the royal victorian eye and ear hospital', `https://www.eyeandear.org.au/page/Research/Research_Committees/`. [Online; accessed March-2019].

Committees, R. (2019), '"honorary researcher information handbook", the royal victorian eye and ear hospital', `https://www.eyeandear.org.au/content/Document/Research/Honorary%20Researcher%20Information%20Handbook%202019%20v6%20Jan19.pdf`. [Online; accessed April-2019].

Dellinger, J. (2018), 'Weight initialisation in neural networks: A journey from the basics to kaiming, towards data science', `http://towardsdatascience.com/weight-initialization-in-neural-networks-a-journey-from-the-basics/`. [Online; accessed September-2019].

Doshi, N. (2018), 'Deep learning best practices - weight initialization, aws, ms data science', http://medium.com/usf-msds/deep-learning-best-practices/. [Online; accessed September-2019].

Fan, R., Chan, T. C., Prakash, G. & Jhanji, V. (2018), 'Applications of corneal topography and tomography: a review', *Clinical & experimental ophthalmology* **46**(2), 133–146.

Fox, M. A., Spicer, K., Chosewood, L. C., Susi, P., Johns, D. O. & Dotson, G. S. (2018), 'Implications of applying cumulative risk assessment to the workplace', *Environment International* **115**, 230–238.

Frucci, M., Riccio, D., Sanniti di Baja, G. & Serino, L. (2016), 'Severe: Segmenting vessels in retina images', *Pattern Recognition Letters* **82**, 162–169.

Gain, P., Jullienne, R., He, Z., Aldossary, M., Acquart, S., Cognasse, F. & Thuret, G. (2016), 'Global Survey of Corneal Transplantation and Eye Banking', *JAMA Ophthalmology* **134**(2), 167–173.

Galvis, V., Sherwin, T., Tello, A., Merayo, J., Barrera, R. & Acera, A. (2015), 'Keratoconus: an inflammatory disorder?', *Eye (London, England)* **29**(7), 843–859.

Gao, Y., Gao, L., Li, X. & Yan, X. (2019), 'A semi-supervised convolutional neural network-based method for steel surface defect recognition', *Robotics and Computer-Integrated Manufacturing* **61**, 101825.

Government, A. (2015), 'National statement on ethical conduct in human research (2007)—updated may 2015', Commonwealth of Australia Canberra, ACT.

Greenwald, M., Scruggs, B., Vislisel, J. & Greiner, M. (2016), 'Corneal imaging: An introduction', http://EyeRounds.org/tutorials/corneal-imaging/index.html/. [Online; accessed September-2019].

Hashemi, H., Beiranvand, A., Yekta, A., Maleki, A., Yazdani, N. & Khabazkhoob, M. (2016), 'Pentacam top indices for diagnosing subclinical and definite keratoconus', *Journal of Current Ophthalmology* **28**(1), 21 – 26.

Heiting, G. (2017), 'All about vision - cornea of the eye', https://www.allaboutvision.com/resources/cornea.htm. [Online; accessed April-2019].

Hossain, M. S., Muhammad, G. & Amin, S. U. (2018), 'Improving consumer satisfaction in smart cities using edge computing and caching: A case study of date fruits classification', *Future Generation Computer Systems* **88**, 333–341.

Jain, A. K., Mao, J. & Mohiuddin, K. (1996), 'Artificial neural networks: A tutorial', *Computer* (3), 31–44.

Joglekar, S. (2017), 'Backpropagation for dummies', `https://codesachin.wordpress.com/2015/12/06/backpropagation-for-dummies/`. [Online; accessed July-2019].

Klein, S. A. (1997), 'Corneal topography reconstruction algorithm that avoids the skew ray ambiguity and the skew ray error.', *Optometry and vision science: official publication of the American Academy of Optometry* **74**(11), 945–962.

Kohavi, R. et al. (1995), A study of cross-validation and bootstrap for accuracy estimation and model selection, *in* 'Ijcai', Vol. 14, Montreal, Canada, pp. 1137–1145.

Kojima, T., Fujimura, S., Hori, R., Okanoue, Y., Shoji, K. & Inoue, M. (2018), 'An innovative voice analyzer "va" smart phone program for quantitative analysis of voice quality', *Journal of Voice* .

Kozak, I., Sample, P. A., Hao, J., Freeman, W. R., Weinreb, R. N., Lee, T.-W. & Goldbaum, M. H. (2007), 'Machine learning classifiers detect subtle field defects in eyes of hiv individuals', *Transactions of the American Ophthalmological Society* **105**, 111–120.

Kymes, S. M., Walline, J. J., Zadnik, K. & Gordon, M. O. (2004), 'Quality of life in keratoconus', *American Journal of Ophthalmology* **138**(4), 527–535.

Kızrak, A. (2019), 'Towards Data Science- Comparison of activation functions for deep neural networks', `https://towardsdatascience.com/comparison-of-activation-functions-for-deep-neural-networks-706ac4284c8a`. [Online; accessed July-2019].

Lavric, A. & Valentin, P. (2019), 'Keratodetect: Keratoconus detection algorithm using convolutional neural networks', *Computational Intelligence and Neuroscience* **2019**, 9.

İlhan, l., Yıldız, b. & Kayrak, M. (2016), 'Development of a wireless blood pressure measuring device with smart mobile device', *Computer Methods and Programs in Biomedicine* **125**, 94–102.

Li, F., Chen, Y., Wang, J., Zhou, X. & Tang, B. (2019), 'A reinforcement learning unit matching recurrent neural network for the state trend prediction of rolling bearings', *Measurement* **145**, 191 – 203.

Lin, X., Wang, X. & Hao, Z. (2017), 'Supervised learning in multilayer spiking neural networks with inner products of spike trains', *Neurocomputing* **237**, 59 – 70.

Ling, S. J., Sanny, J. & Moebs, B. (2019), 'The Law of Reflection, Univeristy of Hawaii', `https://www.cornealdystrophyfoundation.org/what-is-corneal-dystrophy`. [Online; accessed April-2019].

Liu, H., Ning, H., Mu, Q., Zheng, Y., Zeng, J., Yang, L. T., Huang, R. & Ma, J. (2017), 'A review of the smart world', *Future Generation Computer Systems* .

Martin, R. (2018), 'Cornea and anterior eye assessment with slit lamp biomicroscopy, specular microscopy, confocal microscopy, and ultrasound biomicroscopy', *Indian Journal of Ophthalmology* **66**(2), 195–201.

Martínez-Abad, A. & Piñero, D. P. (2017), 'New perspectives on the detection and progression of keratoconus', *Journal of Cataract and Refractive Surgery* **43**(9), 1213 – 1227.

MissingLink (2019), 'Neural Network Concepts - Neural networks for image recognition: Methods, best practices, applications', `https://missinglink.ai/guides/neural-network-concepts/neural-networks-image-recognition-methods-best-practices-applications/`. [Online; accessed June-2019].

Ramachandran, P., Zoph, B. & Le, Q. V. (2017*a*), 'Searching for activation functions', *CoRR* **abs/1710.05941**.

Ramachandran, P., Zoph, B. & Le, Q. V. (2017*b*), 'Swish: a self-gated activation function', *arXiv preprint arXiv:1710.05941* **7**.

Remington, L. A. (2011), *Clinical Anatomy and Physiology of the Visual System, Third Edition*, third edition edn, Butterworth-Heinemann imprint of Elsevier.

Rosa, A. L. B. d. (2013), 'An accessible approach for corneal topography'.

Sadikoglu, F. & Uzelaltinbulat, S. (2016), 'Biometric retina identification based on neural network', *Procedia Computer Science* **102**, 26–33.

Sahebjada, S., Fenwick, E. K., Xie, J., Snibson, G. R., Daniell, M. D. & Baird, P. N. (2014), 'Impact of Keratoconus in the Better Eye and the Worse Eye on Vision-Related Quality of Life', *Investigative Ophthalmology & Visual Science* **55**(1), 412–416.

Sanger, T. D. (1989), 'Optimal unsupervised learning in a single-layer linear feedforward neural network', *Neural Networks* **2**(6), 459 – 473.

Schirmer, K. E. (2003), *Canadian Journal of Ophthalmology* **38**(6), 518.

Shobha, G. & Rangaswamy, S. (2018), *Chapter 8 - Machine Learning*, Vol. 38, Elsevier, pp. 197–228.

Takase, T., Oyama, S. & Kurihara, M. (2018), 'Effective neural network training with adaptive learning rate based on training loss', *Neural Networks* **101**, 68–78.

Tang, R., Fong, S., Deb, S., Vasilakos, A. V. & Millham, R. C. (2018), 'Dynamic group optimisation algorithm for training feed-forward neural networks', *Neurocomputing* **314**, 1–19.

Tur, V. M., MacGregor, C., Jayaswal, R., O'Brart, D. & Maycock, N. (2017), 'A review of keratoconus: Diagnosis, pathophysiology, and genetics', *Survey of Ophthalmology* **62**(6), 770 – 783.

USQ (2018), 'Risk assessment: Risk assessment process', `https://www.usq.edu.au/campus-services/resources/whsmanual/hazsub/hazsub11`. [Online; accessed March-2019].

Vryniotis, V. (2013), 'Tuning the learning rate in gradient descent, datumbox', `http://www.datumbox.com/tuning-the-learning-rate-in-gradient-descent/`. [Online; accessed July-2019].

Wadsworth, J. (2019), 'The corneal dystrophy foundation, what is corneal dystrophy', `https://www.cornealdystrophyfoundation.org/what-is-corneal-dystrophy`. [Online; accessed April-2019].

Wallace, D. A. & Sarver, E. J. (2007), System, Method and Apparatus for Enabling Corneal Topography Mapping by Smartphone, Patent US009839352B2, United States Patent & Trademark Office. `https://pdfpiw.uspto.gov/.piw?Docid=09839352`.

Williams, K. A., Lowe, M. T., Keane, M. C., Jones, V. J., Loh, R. S. & Coster, D. J. (2015), *The Australian corneal graft registry 2015 report.*

Wong, K. C. L., Syeda-Mahmood, T. & Moradi, M. (2018), 'Building medical image classifiers with very limited data using segmentation networks', *Med Image Anal* **49**, 105–116.

Zeinhom, M. M. A., Wang, Y., Sheng, L., Du, D., Li, L., Zhu, M.-J. & Lin, Y. (2018), 'Smart phone based immunosensor coupled with nanoflower signal amplification for rapid detection of salmonella enteritidis in milk, cheese and water', *Sensors and Actuators B: Chemical* **261**, 75–82.

Zimbra, D., Sarangee, K. R. & Jindal, R. P. (2017), 'Movie aspects, tweet metrics, and movie revenues: The influence of ios vs. android', *Decision Support Systems* **102**, 98–109.

Appendix A

Project Specification

For: Sean Thomson

Title: Utilising a Mobile Device to Aid in Early Diagnosis of Corneal Abnormalities

Major: Electrical and Electronic Engineering

Supervisor: John Leis

External Advisor: Dr Srujana Sahebjada – Centre for Eye Research Australia

Enrolment:

ENG4111 – EXT S1, 2019

ENG4112 – EXT S2, 2019

Project Aim:

This project aims to determine if neural networks and a mobile device can be used to help diagnose corneal abnormalities. The hypothesis is that an image taken from a mobile device can be reconstructed into a corneal topography image and can be utilised for automatic diagnosis of abnormalities. This would be without the need for an extensive review from the ophthalmologist or specialised imaging equipment. As corneal abnormalities are quite prevalent in the developing world and frequent access to an ophthalmologist is rare, it would be advantageous if any medical professional could give a preliminary diagnosis and begin treatment or get specialised care for their patient. Additionally, if the specialised equipment is available the automatic analysis by a neural network would enable a preliminary diagnosis to occur quickly enabling extra time to review the patient's history and hereditary relationships.

The conditions and abnormalities that can affect the cornea are extensive. As such, this

project will concentrate on common corneal dystrophies. The most common is keratoconus, Fuchs's dystrophy, lattice dystrophy, and map-dot-fingerprint dystrophy. This project will aim to determine if an automatic diagnosis of keratoconus is feasible using an image from a Pentacam and investigate adapting this for a mobile device. Assessing the feasibility of this automatic diagnosis will indicate implementing similar techniques to diagnose further corneal abnormalities. It will indicate whether new research should be committed to implementing additional features of corneal topography equipment or standalone software that could be used in conjunction with this equipment. Additionally, it will highlight whether it is feasible to create a corneal topographic image from a mobile device.

Programme: Version 2, 20 April 2019

1. Review literature on common corneal dystrophies in particular keratoconus.
2. Review literature on supervised machine learning, neural networks, signal processing, image processing, and eye scanning.
3. Obtain corneal topographic images at various stages of keratoconus and healthy eyes as a control.
4. Sort topographic images into training data and test data and assess suitability.
5. Create a neural network that can be adapted easily as required.
6. Create algorithms for standardising and pre-processing of all images.
7. Test compatibility of the neural network and pre-processing algorithms.
8. Carry out extensive training and testing of the combined algorithm and neural network on keratoconus.
9. Document results based on varying learning rates, training techniques, pre-processing, inputs, number of neurons, number of layers, and epochs.
10. Find optimum solutions for detecting keratoconus from corneal topographic images
11. Document and present results in the form of an academic dissertation.

If successful and time/resources permit:

12. Investigate integrating results from the project for a mobile device.

13. Train a neural network based on optimum solutions to detect Fuch's dystrophy.

14. Document and present results in the form of an academic dissertation.

Appendix B

Risk Assessment

This appendix shows the risk assessment for this project. The risk assessment was initially developed using the project phases of work and utilised the USQ eight-step risk assessment process. The project risk was then summarised with the USQ safety risk management plan. The risk assessment process is discussed in Section 3.14.3.

The USQ risk assessment is shown by Figure B.1, and it was created from Table B.1, B.2, B.3, B.4, and B.5.

Table B.1: Phase 1 - Project risk assessment

Task Number	Risk Description	Risk Score	Mitigation Measures	Final Risk Score
1A	Approval to begin project not given by USQ.	High	Commence early discussions with potential USQ supervisor to ensure the project is appropriate. Ensure approvals are obtained before beginning.	Low
1B	Resources unavailable to commence project.	High	Start resource procurement as soon as the project is approved. If resources cannot be obtained use contingency plan mentioned.	Low
1C	Unable to get permission to utilise current patient's data.	High	Start the permission process as soon as approvals are obtained. Continually communicate with external contacts.	Medium
1D	Unable to gain data from ophthalmologists.	High	Contact several different sources. If data is or becomes unavailable utilise free data available online.	Medium

Table B.2: Phase 2 & 3 - Project risk assessment

Task Number	Risk Description	Risk Score	Mitigation Measures	Final Risk Score
2A	Ethics approvals are rejected by CERA or USQ.	Extreme	Continually communicate with CERA and USQ to ensure approvals are obtained in a timely manner.	Medium
2A	Patient data is accessed by someone not on ethics approvals.	Extreme	Patient data will be received in person not over the internet. Patient data will be stored on encrypted external hard-drive.	Low
2A & 2B	Unable to get significant data from online resources and ophthalmology sources.	Extreme	Ensure there are indicators that there will be enough data before the project is proposed to USQ.	Medium
3A & 3B	MATLAB takes longer than anticipated to train networks.	High	Train some basic networks utilising MATLAB throughout project to determine training time based on available hardware.	Low
3B	Data turns out to not be suitable for training network.	Extreme	Continually seek guidance from literature and USQ supervisor about the quality and amount of data required. Discuss alternative solutions.	Medium

Table B.3: Phase 4 & 5 - Project risk assessment

Task Number	Risk Description	Risk Score	Mitigation Measures	Final Risk Score
Phase 4	Creating pre-processing algorithms is too complex or processing and labelling of images is longer than anticipated.	Extreme	Begin creating pre-processing algorithms before patient data is received to determine potential issues. Ensure enough time and hardware resources are available for image processing.	Medium
5A & 5B	Amending network and training for simplest dataset is more time consuming than anticipated.	High	Start this during Phase 4 as soon as some processed data is available. If needed can run multiple computers concurrently to complete Phases together.	Low
5D	Performance on simple data does not give encouraging results.	High	Make improvements where possible. Investigate why performance is poor and document ways this could be improved for future projects. Still perform Phase 6 to determine results of other input parameters.	Low

Table B.4: Phase 6 & 7 - Project risk assessment

Task Number	Risk Description	Risk Score	Mitigation Measures	Final Risk Score
6A	Time taken to train neural network is longer than anticipated.	High	If there are indicators this is taking too long. There is time within the schedule to extend this task.	Low
Phase 6	The neural network does not perform as expected.	Medium	Document results and make suggestions to how this could be improved. Look into improving this after the conclusion of the project.	Low
7A	Progress report not completed by the deadline or is not enough to continue the project.	High	Ensure this task is taken seriously and began creating it as soon as specifications are released.	Low
7B & 7D	Insufficient time to completed draft dissertation for supervisor feedback. Cannot finish the final dissertation.	High	Start writing a draft dissertation at beginning of the project. Regularly commit blocks of at least 2 hours to work on it. Utilise feedback to complete final copy.	Low
7C	Unable to get leave to present the project at Professional Practice 2.	High	Apply for leave with work at beginning of 2019.	Low

Table B.5: Generic - Project risk assessment

Task Number	Risk Description	Risk Score	Mitigation Measures	Final Risk Score
All Phases	Loss of data or current work due to technical issues.	High	All data will be backed up on a separate hard drive and stored on google drive.	Low
All Phases	Repetitive strain injuries from sitting at a desk and utilising the computer.	High	Take regular breaks when working on the project. Set up workspace ergonomically. Ensure when working on a computer it is in a well-lit area.	Low
All Phases	Can't handle negative results or the stress of large project.	Medium	Carry out activities that promote good mental health which may include exercise and leisure time.	Low

Step 1 (cont)	Step 2	Step 2a	Step 3				Step 4				
Hazards: From step 1 or more if identified	***The Risk:*** What can happen if exposed to the hazard with existing controls in place?	***Existing Controls:*** What are the existing controls that are already in place?	***Risk Assessment:*** Consequence x Probability Risk Level				***Additional controls:*** Enter additional controls if required to reduce the risk level	***Risk assessment with additional controls:***			
			Consequence	**Probability**	**Risk Level**	**ALARP? Yes/no**		**Consequence**	**Probability**	**Risk Level**	**ALARP? Yes/no**
Example											
Working in temperatures over 35º C	Heat stress/heat stroke/exhaustion leading to serious personal injury/death	Regular breaks, chilled water available, loose clothing, fatigue management policy.	catastrophic	possible	high	No	temporary shade shelters, essential tasks only, close supervision, buddy system	catastrophic	unlikely	mod	Yes
Project approval not given.	Cannot complete degree in 2019.		Major	Unlikely	Moderate	No	Commence early discussions with potential USQ supervisor to ensure the project is appropriate. Ensure approvals are obtained before beginning.	Moderate	Rare	Low	No
Resources unavailable.	Can't complete project as planned.		Moderate	Unlikely	Moderate	No	Start resource procurement as soon as the project is approved.	Moderate	Rare	Low	No
Data unavailable.	Project will not be successful.	Search online databases.	Major	Possible	High	No	Start the ethics permission process as soon as possible. Continually monitor ethics approval process.	Major	Unlikely	Moderate	No
Data breach.	Personnel patient information is obtained by third party.		Major	Unlikely	Moderate	No	Patient information will be removed from images. If this is not possible patient data will be received in person not over the internet and data will be stored on encrypted external hard-drive.	Major	Rare	Low	No
Algorithms not successful.	Project objectives will not be met.		Moderate	Possible	High	No	Begin creating preprocessing algorithms before patient data is received to determine potential issues. Ensure enough time and hardware resources are available for image processing.	Moderate	Unlikely	Moderate	No
Insufficient time.	Project objectives will not be met.		Moderate	Unlikely	Moderate	No	Start training and testing as soon as processed data is available. If needed can run multiple computers concurrently to complete Phases together.	Moderate	Rare	Low	No
Methods are unsuccessful.	Project objectives are not met.		Major	Possible	High	No	Make improvements where possible. Investigate why performance is poor and document ways this could be improved for future projects. Still perform Phase 6 to determine results of other input parameters.	Major	Unlikely	Moderate	No
Data loss.	All project progress is lost.		Major	Unlikely	Moderate	No	All data will be backed up on a separate hard drive and stored on google drive.	Major	Rare	Low	No
Repetitive strain injury.	Loss of productivity and student needs treatment for injury.	Use ergonomic workspace.	Moderate	Unlikely	Moderate	No	Take regular breaks when working on the project. Ensure when working on a computer it is in a well-lit area.	Moderate	Rare	Low	No
Stress.	Student may not be able to complete required tasks.		Moderate	Unlikely	Moderate	No	Carry out activities that promote good mental health which may include exercise and leisure time.	Moderate	Rare	Low	No
			Select a consequence	Select a probability	Select a Risk Level	Yes or No		Select a consequence	Select a probability	Select a Risk Level	Yes or No

Figure B.1: USQ Risk Assessment.

Appendix C

Ethical Clearance

This appendix presents the human research ethics approval for this project. It is shown by Figure C.1, and the ethics approval process is further discussed in Section 3.13.

the royal victorian eye and ear hospital

HUMAN RESEARCH ETHICS COMMITTEE
Change of Research Team Member

Note: Form to be submitted per project. If one person in same role being added to multiple projects with the same PI, then can add all on one form (listing all relevant project numbers). If more than one PI, then separate forms are required.

PROJECT NO:	10/954H/15	**PRINCIPAL INVESTIGATOR:**	Prof Paul Baird
PROJECT TITLE:	The Genes in Keratoconus and Myopia Study		
DATE OF ORIGINAL APPROVAL:	01/07/2010	**DATE PROJECT COMMENCED:**	01/07/2010

Are Investigators being added?	YES ✓ NO ☐ If No, please delete this section.
Please complete the following table for each Investigator commencing involvement in the project (please copy and paste extra tables as required)	
Principal Investigator* ☐ Associate Investigator ☐ Contact Person ☐ Student ✓ Other ☐	
* If the Principal Investigator is changing please ensure declaration is counter signed by the current Principal Investigator.	
Title and Full Name	Mr Sean Thomson
Email Address	Student Researcher
Institutional affiliation and position	Centre for Eye Research Australia (External undergraduate student)
Describe research activities responsible for	To undertake data analysis using machine learning algorithms
Describe expertise relevant to the research activities	In final year of Bachelor of Engineering - Electrical and Electronic Major. Completed courses which specialised in digital signal processing and numerous courses involving programming.
Do you require access to Eye and Ear patients or medical records?	YES ☐ NO ✓ If NO, please attach a copy of your CV, and sign declaration below.
Provide details of your current Eye and Ear appointment	☐ Employee ☐ Honorary Researcher - ☐ Approved ☐ Pending ☐ Other
Is the role you will perform for this project within your current approved scope of practice (SOP)?	YES ✓ NO ☐
Provide details of your current appointment	*Role External student currently undertaking undergraduate dissertation investigating automatic detection of Keratoconus. Co-Supervisor is Dr Srujana Sahebiada.*

Research Team Member declaration

I certify that:

- All information in this application and supporting documentation is correct and as complete as possible;
- I have read and understand the application and protocol and HREC approval conditions;
- I have familiarised myself with, considered and addressed in this application any relevant legislation, regulations, research guidelines and organisational policies;
- All relevant financial and non-financial interests of the project team have been disclosed; and
- In the capacity of a supervisor, as applicable, I have reviewed this application and I will provide appropriate supervision to the student(s) in accordance with the arrangements specified in this application and those associated with the student's educational program.

Signature: Sean Thomson Date: 27 Jun 19

Does the Participant Information & Consent Form (PICF) need updating with the research team changes? Only the name of the Principal Investigator and Contact Person should appear in the PICF. Should either change, an updated PICF should be submitted with this form. If any other changes, please submit an amendment form separately.	YES ☐ NO ✓ If Yes, attach updated version and list below: PICF Name, Version _____ Date _____

Other Relevant Information
Please add any other information that will assist the Research Office in reviewing this change to research team member.

Principal Investigator declaration
I agree to add the above listed researcher/s to the research team, and confirm that they have the training and experience required to complete their role in this project.

Principal Investigator Signature: Date: 27th June 2019

Form Version 5 March 2019

Office Use Only			
SOP = Role ☑	Approved ☑	Not approved ☐	HREC approval signature:

This decision will be ratified by the full Human Research Ethics Committee at the next meeting scheduled. This amendment can proceed from the date listed below on this form. No further correspondence will be provided on this amendment unless the HREC requests additional information.

Database updated ☑

Name: KERRYN BAKER Signed: K Baker Date: 27/6/19

Figure C.1: Human Research Ethics Approval.

Appendix D

Project Schedule

This appendix shows the project schedule. Figure D.1 represents Semester 1 and 2 as weeks. Week 1 being the beginning of the project on the 25th of February 2019. The start and end time for each task are shown using a horizontal blue bar. Tasks that do not begin until the conclusion of another are dependent on these previous tasks.

The project schedule and phases of work are discussed in Section 3.14.2

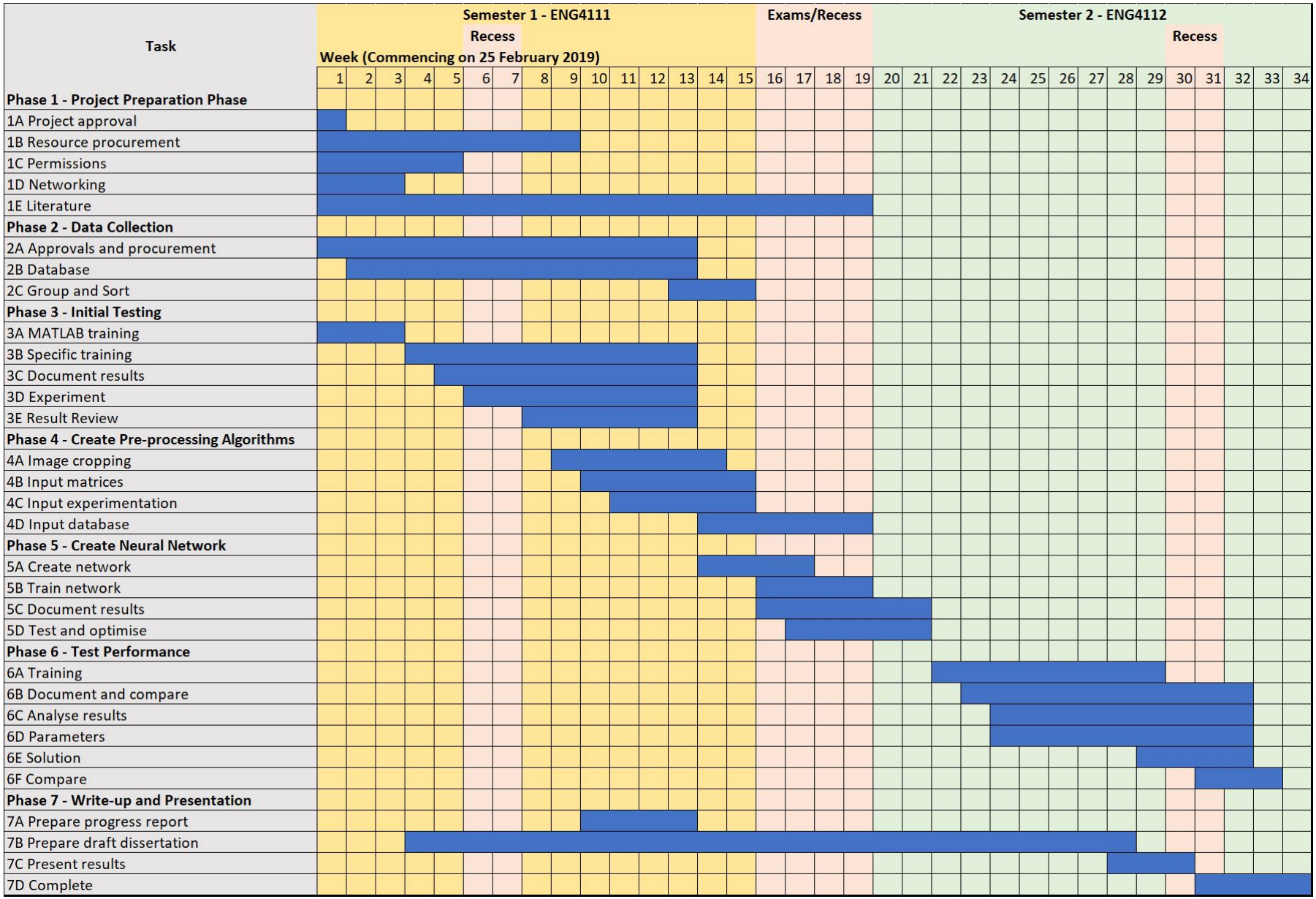


Figure D.1: Project Schedule

Appendix E

Honorary Researcher Application

This appendix shows the Honorary Researcher Application submitted to the Royal Victorian Eye and Ear Hospital. This application process is discussed in Section 3.13.1.

HONORARY RESEARCHER APPLICATION

THIS FORM MUST BE COMPLETED FOR NEW RESEARCHERS WHO DO NOT HAVE AN EYE AND EAR APPOINTMENT AND REQUIRE ACCESS TO PATIENTS OR MEDICAL RECORDS
OR
FOR HONORARY RESEARCHERS WHO WISH TO AMEND THEIR APPROVED SCOPE OF PRACTICE

One signed form to be submitted prior to commencement on research project at the Eye and Ear Hospital. Once an Eye and Ear appointment has been granted, you may also need to submit a new HREC application or an HREC Change of a Research Team Member form requesting new personnel to be added to a project.

APPLICANT'S DETAILS

Title	Given Name	Surname	Email	Phone
Mr	Sean	Thomson	Sean_thomson@live.com.au	0401516088

Institutional affiliation and position	Centre for Eye Research Australia. External Undergraduate Student.

Do you have a current Eye and Ear honorary researcher appointment?	Yes ☐ No ☒
	If Yes, please provide details of current SOP and requested SOP. Current SOP: Requested SOP:
	Please provide justification for the change in your current appointment.

PURPOSE OF APPOINTMENT

Employee of research partner or external organisation to be added to a research project team ☐	If yes, is it a ☐ current approved project; or ☐ project to be submitted in future *Specify the HREC reference number and title*
Medical Student involved in a clinical audit / research project / research unit ☐	If yes, specify name of Eye and Ear Supervisor Note: medical students can only have a Scope of Practice for access to medical records, as all other procedures are done under supervision of their Supervisor.
Previous Eye and Ear employee requiring Honorary appointment to continue role on a clinical audit / research project ☐	If yes, specify the HREC reference number and title
Other ☒	Please provide details External student currently undertaking undergraduate dissertation investigating automatic detection of corneal dystrophies. Advisor is Dr Srujana Sahebjada.

What duration of appointment is required?	Please provide timeframe and justification Until November 2019. That will be the conclusion of my project.

GOOD CLINICAL PRACTICE TRAINING

Have you completed Good Clinical Practice (GCP) TransCelerate approved training?	Yes ☐ No ☒ If YES, please provide a copy of the current certificate of training.

QUALIFICATIONS

	Year	Qualification	University/College*
Graduation:	End 2019	Bachelor of Engineering (Honours)	University of Southern Queensland
Postgraduate:			
Profession/Course:		**Electrical and Electronic Major**	
Registration:	AHPRA: Yes ☐ No ☒ AHPRA Reg No: Other:		

ENGLISH LANGUAGE PROFICIENCY

Specify which AHPRA* English language skills and registration standards apply	☒ English language skills registration standard ☐ English language skills registration standard (Aboriginal and Torres Strait Islander health) ☐ English language skills registration standard (medical) ☐ English language skills registration standard (2019) (nursing and midwifery) ☐ Other – please specify
Specify which pathway applies	☒ English is your primary language ☐ Secondary education and tertiary qualifications in a recognised country ☐ Six years of continuous education in a recognised country ☐ Achieved minimum required Test results (IELTS / OET / PTE / TOEFL) ☐ Other

*Australian Health Practitioner Regulation Agency (AHPRA)

ROLE IN PROJECT

Please refer to the matrix on page 4 and attach the evidence/certificates required for the requested role.

What Scope of Practice do you require for this role? (please add/delete as appropriate)

Please refer to the Verified SOPs (VSOP) on the Research website to ensure that you meet the required qualification and experience.

Verified SOPs
- Access to medical record (including DHR)
- Conduct ECGs (VSOP)
- Orthoptist (VSOP)
- Patient Recruitment and obtaining consent, conducting interviews/questionnaires (VSOP)
- Principal Investigator with oversight of research project but no patient contact at or on behalf of the Eye and Ear Hospital (VSOP)
- Venepuncture (VSOP)

Other roles
- Assessing vital signs
- Audiologist
- Conduct ERGs
- Conduct retinal imaging (including OCT, fundus photography)
- Medical Officer
- Obtain blood samples and skin biopsy
- Optometrist
- Slit lamp examination (not including administering eye drops)
- Speech Pathologist
- Taking blood pressures
- Physiotherapist
- Visual acuity measurements

Other:

If you require access to medical record (including DHR), please specify which program/system you need access to:

Clinical Systems	Yes	☐	No	☒
PiMS	Yes	☒	No	☐
IFA	Yes	☐	No	☒
DHR	Yes	☐	No	☒

What training have you received for this role? (please provide evidence of training)

Will be under supervision from CERA and have carried out Health Record Act Training.

If you require extra training to perform the roles listed above, please detail who will provide the training.

Other Relevant Information for the project
Please add any other information that will assist the Research Office in reviewing this application.

Declaration by Applicant

I agree:

- to keep confidential all information and data that relates to individuals involved in research/audit projects. I shall not make any direct copy of participant's records;
- to keep confidential any information concerning persons or events that comes to my attention at The Royal Victorian Eye and Ear Hospital. Such information includes anything relating to the project/audit above, and any other information which I hear, see or read during my time at the hospital;
- to only use data and any tissue samples collected for the study for which approval has been given;
- to maintain security procedures for the protection of privacy.
- that I have read the Eye and Ear Hospital's Researcher Handbook and agree to comply.

Please tick the following if appropriate:

☒ **I have read the NH&MRC *National Statement on Ethical Conduct in Human Research (*2007) and will observe the principles set out in that document and in the *Declaration of Helsinki* and ICH Good Clinical Practice.**

Applicant's Signature: Date:

Eye and Ear Hospital Authorisation

Principal Investigator (if EAE employed)
Or
Head of Unit/Clinic Signature: Date:

Please submit the completed application form, along with the supporting documentation (tick below as appropriate – as per the Research Training Matrix on page 4 of this form) to the Research Office (ethics@eyeandear.org.au):

☒ Curriculum Vitae (CV)
☒ a recent (no older than 3 months) National Police Check
☐ Working With Children Check
☒ Health Record Act certificate
☐ Hand Hygiene certificate
☐ Good Clinical Practice certificate

RESEARCHER TRAINING MATRIX

Role in Project \ Credentialing Requirement	Researcher Declaration Form	CV[1]	Police Check[2]	WWCC[3]	Health Record Act[4] (if not AHPRA registered)	Hand Hygiene[5]	Good Clinical Practice[6]
Eye and Ear employed researcher	√	√	In place as employee	In place as employee, if working with children in scope of role	In place as employee	In place as employee	Reqd if Principal Investigator
Principal Investigator on interventional drug/device trial	√	√	√	If working with children in project scope and contact with a child is unsupervised, direct and a part of the person's duties	√	√	√
Principal Investigator on non-interventional clinical research project with patient contact and medical record access	√	√	√	If working with children in project scope and contact with a child is unsupervised, direct and a part of the person's duties	√	√	√
Principal Investigator with oversight of research project with no Eye and Ear patient contact	√	√	√	-	√	-	√
Researcher on project with patient contact and medical record access	√	√	√	If working with children in project scope and contact with a child is unsupervised, direct and a part of the person's duties	√	√	Not at this point but highly recommended
Researcher on project involving medical record access only	√	√	√	-	√	-	Not at this point but highly recommended
Researcher on project with no Eye and Ear patient contact and no medical record access	√	√	-	-	-	-	Not at this point but highly recommended

[1] Curriculum Vitae

[2] National Police Check – name only check required http://www.police.vic.gov.au/content.asp?Document_ID=274

[3] WWCC - If working with children in project scope and contact with a child is unsupervised, direct and a part of the person's duties http://www.workingwithchildren.vic.gov.au/home/applications/

[4] Health Record Act - https://ohsc.e3learning.com.au/

[5] Hand Hygiene - http://www.hha.org.au

[6] GCP - Highly recommended http://www.transceleratebiopharmainc.com/gcp-training-attestation/ (for list of approved providers of GCP training)

Figure E.1: Honorary Researcher Application.

Appendix F

Health Record Act Training

This appendix shows evidence of Health Records Act Training completed for this research. The course was developed from the Health Complaints Commissioner to give researchers an understanding of the Health Records Act (2001). This training is further discussed in Section 3.13.3.

Certificate of Completion

THIS IS TO CERTIFY THAT

Sean Thomson

HAS SUCCESSFULLY COMPLETED

Health Records Act

AND THIS COURSE WAS COMPLETED ON

29 May 2019

HEALTH COMPLAINTS COMMISSIONER

Figure F.1: Health Records Act Training

Appendix G

Cropping Function

This appendix shows the cropping function created for this project with MATLAB. It receives a Pentacam image, removes irrelevant information and will output four images. This function is discussed in Section 4.2.1.

Listing G.1: Cropping Function.

```
function [axial, thickness, elevFront, elevBack] =
    ↪ refractiveMapCrop(image)
%[axial, thickness, elevFront, elevBack] = refractiveMapCrop(
    ↪ image)
%   This function will take a Pentacam 4 Maps Refractive image
    ↪ and split it into each of the 4 topographic outputs (
    ↪ Sagittal Curvature, Corneal, Thickness, Elevation front
    ↪ and back).
%   The 4 outputs are square, RGB images with all other data
    ↪ discriminated.


%Variables
square = [286 286]; % width and height. Ensure square for neural
    ↪  network
axialMin =[450 132];
corneaMin = [450 509];
elevFrontMin = [825 132];
elevBackMin = [825 509];

circDim = [145 145 142]; % centre and radius of circle [row col
    ↪ r]

% Split image into each individual segment and ensure image is
    ↪ square
aImage = imcrop(image, [axialMin square]);
tImage = imcrop(image, [corneaMin square]);
```

```
efImage = imcrop(image, [elevFrontMin square]);
ebImage = imcrop(image, [elevBackMin square]);

%figure(2)
%imshow(aImage) % show image for testing

% Convert to 2D to create mask for cropping of circle
imageGray = rgb2gray(aImage);
imageSize = size(imageGray);

% Create circle and find points
[x,y] = ndgrid((1:imageSize(1))-circDim(1),(1:imageSize(2))-
    ↪ circDim(2));
mask = uint8((x.^2 + y.^2) < circDim(3)^2);

%-----------------------------
% Used to see visible circle for fine adjustments as required
% h = viscircles(circDim(1:2),circDim(3));
% c = h.Children(1).XData(1:end-1);
% r = h.Children(2).YData(1:end-1);

% Create grid to be used for mask
%[C, R] = meshgrid(1:imageSize(2),1:imageSize(1));

% Find points inside of the circle
%mask = inpolygon(R,C,y,x);
%-----------------------------

% Discriminate area not within ROI and reconstruct all images
axial = uint8(zeros(imageSize));
axial(:,:,[1 2 3]) = aImage(:,:,[1 2 3]).*uint8(mask);

thickness = uint8(zeros(imageSize));
thickness(:,:,[1 2 3]) = tImage(:,:,[1 2 3]).*uint8(mask);

elevFront = uint8(zeros(imageSize));
elevFront(:,:,[1 2 3]) = efImage(:,:,[1 2 3]).*uint8(mask);

elevBack = uint8(zeros(imageSize));
elevBack(:,:,[1 2 3]) = ebImage(:,:,[1 2 3]).*uint8(mask);

end
```

Appendix H

Dimension Reduction and Sorting

This appendix shows the dimension reduction and sorting program created for this project with MATLAB. This program reduces three-dimensional images to two-dimensions and saves them in appropriate folders to be accessed by other algorithms. This program is discussed in Section 4.2.2.

Listing H.1: Dimension Reduction and Sorting.

```
%% ENG49112 Research Project
% Pre-processing Training Data
% Author:    Sean Thomson
% Date:      27Jul19
% Version:   5
% Notes:     Created function to call cropping instead of having
    ↪ it all in the same script.
%            This script is for initially processing training
    ↪ images.
%            Make images invariant to RGB dimensions.
%            Create folders for separate RGB and greyscale
    ↪ channels to determine best results.

% Initialise
clc, clear, close all

% Folder names for processed data.
controlF = "control_images_processed_RGB";
KCNF = "KCN_processed_RGB";
controlFG = "control_images_processed_channels";
KCNFG = "KCN_processed_channels";
axialF = "axial";
thicknessF = "thickness";
elevationFrontF = "elevation_front";
elevationBackF = "elevation_back";
rChannel = "R";
```

```
gChannel = "G";
bChannel = "B";
grayChannel = "Gray";

% Check for folder and create if not present in current MATLAB
    ↪ directory.
mainFolders = [controlF, KCNF, controlFG, KCNFG];
subFolders = [axialF, thicknessF, elevationFrontF,
    ↪ elevationBackF];
subSubFolders = [rChannel gChannel bChannel grayChannel];

for i = 1:length(mainFolders)

    for j = 1:length(subFolders)

        for k = 1:length(subSubFolders)

            if ~exist(mainFolders(i), 'dir')
                mkdir(mainFolders(i))
            end

            currentSub = strcat(mainFolders(i),'\',subFolders(j)
                ↪ );

            if ~exist(currentSub, 'dir')
                mkdir(currentSub)
            end

            currentSub2 = strcat(mainFolders(i),'\',subFolders(j
                ↪ ),'\',subSubFolders(k));

            if ~exist(currentSub2, 'dir')
                mkdir(currentSub2)
            end
        end
    end
end


% Get a list of all .jpg in control images folder. This only
    ↪ works if
% images folder is in current MATLAB directory.
imageFiles = dir('control_images\*.jpg');
nFiles = length(imageFiles);    % Number of files found

% Loop to process all control images and save in new location
for i = 1:nFiles
    currentFileName = imageFiles(i).name;
    fileName = ['control_images\' currentFileName];

    % Read Image
    image = imread(fileName);
```

```
    % Call refractive map crop function
    [axial, thickness, elevFront, elevBack] = refractiveMapCrop(
        ↪ image);

    % Save for future use in appropriate folder
    imwrite(axial,strcat(mainFolders(1), '\', axialF, '\','
        ↪ axial_control_', currentFileName))
    imwrite(thickness,strcat(mainFolders(1), '\', thicknessF, '\
        ↪ ','thickness_control_', currentFileName))
    imwrite(elevFront,strcat(mainFolders(1), '\',
        ↪ elevationFrontF, '\','elevFront_control_',
        ↪ currentFileName))
    imwrite(elevBack,strcat(mainFolders(1), '\', elevationBackF,
        ↪  '\','elevBack_control_', currentFileName))

    % Loop for separate channels
    for j = 1:length(subSubFolders)
        if j < 4
            % Convert images to separate channels for
                ↪ dimensionality
            % reduction
            axialC = (axial(:,:,j));
            thicknessC = (thickness(:,:,j));
            elevFrontC = (elevFront(:,:,j));
            elevBackC = (elevBack(:,:,j));
        else
            axialC = rgb2gray(axial);
            thicknessC = rgb2gray(thickness);
            elevFrontC = rgb2gray(elevFront);
            elevBackC = rgb2gray(elevBack);
        end

        % Save for future use in appropriate folder
        imwrite(axialC,strcat(mainFolders(3), '\', axialF, '\',
            ↪ subSubFolders(j),'\axial_control_channel_',
            ↪ currentFileName))
        imwrite(thicknessC,strcat(mainFolders(3), '\',
            ↪ thicknessF, '\',subSubFolders(j),'\
            ↪ thickness_control_channel_', currentFileName))
        imwrite(elevFrontC,strcat(mainFolders(3), '\',
            ↪ elevationFrontF, '\',subSubFolders(j),'\
            ↪ elevFront_control_channel_', currentFileName))
        imwrite(elevBackC,strcat(mainFolders(3), '\',
            ↪ elevationBackF, '\',subSubFolders(j),'\
            ↪ elevBack_control_channel_', currentFileName))
    end
end

% Get a list of all .jpg in control images folder. This only
    ↪ works if
% images folder is in current MATLAB directory.
```

```
imageFilesKCN = dir('kc_images\*.jpg');
nFiles = length(imageFilesKCN);     % Number of files found

% Loop to process all KCN images and save in new location
for i = 1:nFiles
    currentFileName = imageFilesKCN(i).name;
    fileName = ['kc_images\' currentFileName];

    % Read Image
    image = imread(fileName);

    % Call refractive map crop function
    [axial, thickness, elevFront, elevBack] = refractiveMapCrop(
        ↪ image);

    % Save for future use in appropriate folder
    imwrite(axial,strcat(mainFolders(2), '\', axialF, '\','
        ↪ axial_KCN_', currentFileName))
    imwrite(thickness,strcat(mainFolders(2), '\', thicknessF, '\
        ↪ ','thickness_KCN_', currentFileName))
    imwrite(elevFront,strcat(mainFolders(2), '\',
        ↪ elevationFrontF, '\','elevFront_KCN_', currentFileName
        ↪ ))
    imwrite(elevBack,strcat(mainFolders(2), '\', elevationBackF,
        ↪  '\','elevBack_KCN_', currentFileName))

    % Loop for separate channels
    for j = 1:length(subSubFolders)
        if j < 4
            % Convert images to separate channels for
                ↪ dimensionality
            % reduction
            axialC = (axial(:,:,j));
            thicknessC = (thickness(:,:,j));
            elevFrontC = (elevFront(:,:,j));
            elevBackC = (elevBack(:,:,j));
        else
            axialC = rgb2gray(axial);
            thicknessC = rgb2gray(thickness);
            elevFrontC = rgb2gray(elevFront);
            elevBackC = rgb2gray(elevBack);
        end

        % Save for future use in appropriate folder.
        imwrite(axialC,strcat(mainFolders(4), '\', axialF, '\',
            ↪ subSubFolders(j),'\axial_KCN_channel_',
            ↪ currentFileName))
        imwrite(thicknessC,strcat(mainFolders(4), '\',
            ↪ thicknessF, '\',subSubFolders(j),'\
            ↪ thickness_KCN_channel_', currentFileName))
        imwrite(elevFrontC,strcat(mainFolders(4), '\',
            ↪ elevationFrontF, '\',subSubFolders(j),'\
```

```
            ↪ elevFront_KCN_channel_', currentFileName))
        imwrite(elevBackC, strcat(mainFolders(4), '\',
            ↪ elevationBackF, '\',subSubFolders(j),'\
            ↪ elevBack_KCN_channel_', currentFileName))
    end
end

% Used to show saved images. Can be used for debugging.
% I = double(test);
% X = reshape(I,size(I,1)*size(I,2),3);
% coeffKCN = pca(X);
% Itransformed = X*coeff;
% Ipc1 = reshape(Itransformed(:,1),size(I,1),size(I,2));
% Ipc2 = reshape(Itransformed(:,2),size(I,1),size(I,2));
% Ipc3 = reshape(Itransformed(:,3),size(I,1),size(I,2));
% figure, imshow(Ipc1,[]);
% figure, imshow(Ipc2,[]);
% figure, imshow(Ipc3,[]);
```

Appendix I

Data Augmentation and Scaling

This appendix shows the data augmentation and scaling program created for this project with MATLAB. This program augments and prepares the data for training with an ANN and saves images in the training folder. This program is discussed in Section 4.2.3.

Listing I.1: Data Augmentation and Scaling.

```
%% ENG49112 Research Project
% Data Augmentation
% Author:    Sean Thomson
% Date:      29Jul19
% Version:   4
% Notes:     This is to run after pre-processing cropping and
    ↪ will create several variations of the saved images.
%            They will then be collected in a master folder ready
    ↪  to input to a neural network.
%            Changed due to separate RGB and grey channels of
    ↪ data from pre-processing algorithm.
%            Changed labelling convention for testing in
    ↪ neuralInputV9
%            Fixed labelling bug where it was not incremental

% Initialise
clc, clear, close all

% Variables
noRotations = 4;
reduceFactor = .1;

% Folder names for processed data.
imageFolderKCN = "KCN_processed_channels";
imageFolderCont = "control_images_processed_channels";
trainDataF = "Training_data_channel_reduced";
axialF = "axial";
```

```
thicknessF = "thickness";
elevationFrontF = "elevation_front";
elevationBackF = "elevation_back";
dataLabel = ["control" "KCN"];
rChannel = "R";
gChannel = "G";
bChannel = "B";
grayChannel = "Gray";

% Folder arrays
imageFolder = [imageFolderCont imageFolderKCN];
subFolders = [axialF, thicknessF, elevationFrontF,
    ↪ elevationBackF];
subSubFolders = [rChannel gChannel bChannel grayChannel];

% Check for the folder and create if not present in current
    ↪ MATLAB directory

        if ~exist(trainDataF, 'dir')
            mkdir(trainDataF)
        end

% Loop to create subfolders
    for j = 1:length(subFolders)

        for k = 1:length(subSubFolders)

            currentSub = strcat(trainDataF,'\',subFolders(j));

            if ~exist(currentSub, 'dir')
                mkdir(currentSub)
            end

             currentSub2 = strcat(trainDataF,'\',subFolders(j)
                ↪ ,...
                 '\',subSubFolders(k));

            if ~exist(currentSub2, 'dir')
                mkdir(currentSub2)
            end
        end
    end



% Nested loop to get all previously processed images reduce them
    ↪ in size,
% rotate each image and save in a master folder.
for k = 1:length(imageFolder)

    for h = 1:length(subFolders)
```

```
for m = 1:length(subSubFolders)
    % Get a list of all .jpg in control images folder.
       ↪ This only
    % works if images folder is in current MATLAB
       ↪ directory.
    imageFiles = dir(strcat(imageFolder(k),'\',
       ↪ subFolders(h),'\',subSubFolders(m),'\*.jpg'));
    nFiles = length(imageFiles);     % Number of files
       ↪ found

    % Conditional test to ensure all images are
       ↪ sequential
    if k < 2
        labelAdd = nFiles;
        label = 1; % variable to label each saved image
    else
        label = labelAdd + 1;
    end

    % Loop to process all control images and save in new
       ↪  location
    for i = 1:nFiles
        currentFileName = imageFiles(i).name;
        fileName = strcat(imageFolder(k),'\',subFolders(
           ↪ h),'\',subSubFolders(m),'\',
           ↪ currentFileName);

        % Read Image
        image = imread(fileName);
        image = imresize(image,reduceFactor);

        for j = 1:noRotations

            % Rotate each image and save each rotation
            currentAngle = 360/noRotations*j;
            imageR = imrotate(image,currentAngle);
            if label < 10
             % Save for future use in appropriate folder
                imwrite(imageR,strcat(trainDataF, '\',
                   ↪ subFolders(h),'\', subSubFolders(m
                   ↪ ),'\',dataLabel(k),'_0',num2str(
                   ↪ label),'_',num2str(currentAngle),'
                   ↪ .jpg'));
            else
                imwrite(imageR,strcat(trainDataF, '\',
                   ↪ subFolders(h),'\',subSubFolders(m)
                   ↪ ,'\',dataLabel(k),'_',num2str(
                   ↪ label),'_',num2str(currentAngle),'
                   ↪ .jpg'));
            end

        end
```

```
                % Increment outside of rotations to determine
                    ↪ image sets
                label = label + 1;
            end
        end
    end
end
```

Appendix J

Forward Propagation Function

This appendix shows the forward propagation function created for this project with MATLAB. It calculates one complete forward pass of the ANN and will output the prediction and result of each layer. This function is discussed in Section 4.3.1.

Listing J.1: Forward Propagation Function.

```
function [hiddenOutput,hiddenHiddenIIOutput,
    ↪ hiddenIIHiddenIIIOutput,y] = fwdPropagationSigmoid3(
    ↪ ihWeights,hhIIWeights,hIIhIIIWeights,hoWeights,hiddenBias,
    ↪ hiddenIIBias,hiddenIIIBias,outputBias,x)
%[hiddenOutput,hiddenHiddenIIOutput,hiddenIIHiddenIIIOutput,y] =
    ↪  fwdPropagationSigmoid3(ihWeights,hhIIWeights,
    ↪ hIIhIIIWeights,hoWeights,hiddenBias,hiddenIIBias,
    ↪ hiddenIIIBias,outputBias,x)
%  This will calculate the forward propagation of a neural
    ↪ network using the sigmoid activation functions. It will
    ↪ only accept one hidden layer.
%  The inputs required are input to hidden weights, hidden to
    ↪ output weights, hidden bias, output bias, and the
    ↪ normalised input.
%  It will output both the hidden and final output.

        % Calculate forward propagation
        % Output for each hidden neuron normalised using sigmoid
            ↪  function
        hiddenOutput = 1./(1 + exp(-(ihWeights*x) + hiddenBias))
            ↪ ;
        % Output for hidden hiddenII neurons normalised using
            ↪ sigmoid
        hiddenHiddenIIOutput = 1./(1+exp(-(hhIIWeights*
            ↪ hiddenOutput)+ hiddenIIBias));
        % Output for hiddenII to hiddenIII normalised with
            ↪ sigmoid
```

```
        hiddenIIHiddenIIIOutput = 1./(1+exp(-(hIIhIIIWeights*
            ↪ hiddenHiddenIIOutput) + hiddenIIIBias));

        % Output normalised using sigmoid function
        y = 1./(1 + exp(-(hoWeights*hiddenIIHiddenIIIOutput) +
            ↪ outputBias));
end
```

Appendix K

Individual Topographic Map Training

This appendix shows the topographic map training program created for this project with MATLAB. This program can train each ANN using either the red, green, or blue RGB channel. This choice can be modified by changing variables before execution. The classifier is saved for future access after training. This program is discussed in Section 4.3.2.

Listing K.1: Individual Topographic Map Training.

```
%% ENG49112 Research Project
% Final Model Training Topographic Data
% Author: Sean Thomson
% Date: 22SEP19
% Version: 2
% Version Note: Adapted from neuralInputV12
%               Initialise biases to zero

% Initialise
clc, clear, close all
iniTime = clock;

% Folder names for processed data.
trainDataF = 'Training_data_channel_reduced';
axialF = 'axial';
thicknessF = 'thickness';
elevationFrontF = 'elevation_front';
elevationBackF = 'elevation_back';
dataLabel = ['control' 'KCN'];      % naming convention of
    ↪ images
rChannel = 'R';
gChannel = 'G';
```

```
bChannel = "B";
grayChannel = "Gray";
subFolders = [axialF, thicknessF, elevationFrontF,
    ↪ elevationBackF];
subSubFolders = [rChannel gChannel bChannel];

% Channels based on testing
axialChannel = gChannel;
thicknessChannel = gChannel;
elevationFChannel = bChannel;
elevationBChannel = gChannel;
currentChannel = [axialChannel, thicknessChannel,
    ↪ elevationFChannel,...
     elevationBChannel];

% Loop to compare all parameters
for m = 1:length(subFolders)
    % Determine total number of images in training folder and
        ↪ pixel count of images
    totalFiles = dir(strcat(trainDataF,'\',subFolders(m),'\',
        ↪ currentChannel(m),'\*.jpg'));
    nFilesTotal = length(totalFiles);     % Number of files found
    currentFileName = totalFiles(1).name;
    fileName = strcat(trainDataF,'\',subFolders(m),'\',
        ↪ currentChannel(m),'\',currentFileName);
    imageCount = imread(fileName);
    pixelNumber = numel(imageCount);

    % Allocate matrix before loops
    image = zeros(pixelNumber+3,1);
    axialMatrix = zeros(pixelNumber+3,nFilesTotal);
    nCount = nFilesTotal;

    for j = 1:length(dataLabel)
        % Get a list of all .jpg in control images folder. This
            ↪ only works if images folder is in current MATLAB
            ↪ directory.
        imageFiles = dir(strcat(trainDataF,'\',subFolders(m),'\'
            ↪ ,...
             currentChannel(m),'\*',dataLabel(j),'*.jpg'));
        nFiles = length(imageFiles);     % Number of files found

        for i = 1:nFiles
            currentFileName = imageFiles(i).name;
            fileName = strcat(trainDataF,'\',subFolders(m),'\'
                ↪ ,...
                 currentChannel(m),'\',currentFileName);

            % Create label at top of matrix to remove an input
                ↪ for testing and training
            numOut = regexp(currentFileName,'\d+','match');%find
                ↪  integers in name
```

```
            numOut = numOut(1); %exclude angle
            numOut = str2double(cat(1,numOut{:})); %string to
                ↪ integer

            % Read image and convert to single and normalise to
                ↪ 2 to -2
            imageRead = imread(fileName);
            imageRead = im2single(imageRead);
            imageRead = (imageRead.*4) - 2;
            imageRead = imageRead(:);
            imageRead = [numOut; imageRead];

            % Label data within matrix a 1 in the last row
                ↪ indicates a
            % patient with KCN
            if dataLabel(j) == "KCN"
                image = [imageRead; 1];
                image = [image; 0];
            else
                image = [imageRead; 0];
                image = [image; 1];
            end
            axialMatrix(:,nCount) = image;
            nCount = nCount - 1;
        end
    end
    % Switch statement to create input matrix for each channel
    switch subFolders(m)
        case axialF
            axialMatrixFM = axialMatrix;
        case thicknessF
            thicknessMatrixFM = axialMatrix;
        case elevationFrontF
            elevationFrontMatrixFM = axialMatrix;
        case elevationBackF
            elevationBackMatrixFM = axialMatrix;
    end
end

%% Neural network
% Variables
noInput = pixelNumber;
noHidden = round(2*noInput);
noHiddenII = round(2/3*noHidden);
noHiddenIII = round(2/3*noHiddenII);
noOutput = 2;
lr = 0.001; %learning rate
noEpoch = 600;
costPlot = gpuArray(zeros(noEpoch,length(subFolders)));

% Loop to train all parameters separately
for para = 1:length(subFolders)
```

```
% Pick relevant data for training and set file name for
    ↪ saving
% weights and biases
switch subFolders(para)
    case axialF
        axialMatrix = axialMatrixFM;
        fileName = 'axialMatrices\';
    case thicknessF
        axialMatrix = thicknessMatrixFM;
        fileName = 'thicknessMatrices\';
    case elevationFrontF
        axialMatrix = elevationFrontMatrixFM;
        fileName = 'elevationFrontMatrices\';
    case elevationBackF
        axialMatrix = elevationBackMatrixFM;
        fileName = 'elevationBackMatrices\';
end

% Ensure weights are comparative to network. This uses
    ↪ Xavier Initialisation technique.
weightSetI = sqrt(2/(noInput+noHidden));
weightSetHII = sqrt(2/(noHidden+noHiddenII));
weightSetHIII = sqrt(2/(noHiddenII+noHiddenIII));
weightSetO = sqrt(2/(noHiddenIII+noOutput));

% Weights from input to hidden layer
ihWeights = gpuArray((rand(noHidden,noInput).*weightSetI) -
    ↪ (weightSetI/2));
% Weights from hidden to hiddenII layer
hhIIWeights = gpuArray((rand(noHiddenII,noHidden).*
    ↪ weightSetHII) - (weightSetHII/2));
% Weights from hiddenII to hiddenIII layer
hIIhIIIWeights = gpuArray((rand(noHiddenIII,noHiddenII).* (
    ↪ weightSetHIII) - (weightSetHIII/2));
% Weights from hidden layer to output
hoWeights = gpuArray((rand(noOutput,noHiddenIII).*weightSetO
    ↪ ) -(weightSetO/2));
% Bias initialsed to zero
hiddenBias = gpuArray(zeros(noHidden,1));
hiddenIIBias = gpuArray(zeros(noHiddenII,1));
hiddenIIIBias = gpuArray(zeros(noHiddenIII,1));
outputBias = gpuArray(zeros(noOutput,1));

for i = 1:noEpoch

    MSE = 0;
    % Create pointer array for image selection
    trainPointer = 1:nFilesTotal;

    % Epoch loop to get random inputs
    for j = 1:length(trainPointer)
```

```
% Find random index then use index to point to next
    ↪ propagation
randomIndex = randi(length(trainPointer),1);
x = gpuArray(axialMatrix(2:end-2,trainPointer(
    ↪ randomIndex)));
desiredResult = [axialMatrix(end-1,trainPointer(
    ↪ randomIndex));axialMatrix(end,trainPointer(
    ↪ randomIndex))];

% Calculate forward propogation
[hiddenOutput,hiddenHiddenIIOutput,
    ↪ hiddenIIHiddenIIIOutput,y] =
    ↪ fwdPropagationSigmoid3(ihWeights,hhIIWeights,
    ↪ hIIhIIIWeights,hoWeights,hiddenBias,
    ↪ hiddenIIBias,hiddenIIIBias,outputBias,x);

% Calculate error, error = target - output
outputError = desiredResult - y;
MSE = MSE + (outputError.^2);

% Calculate hidden III error to output, error =
    ↪ output_error * h_o_weights
hiddenIIHiddenIIIError = (transpose(hoWeights) *
    ↪ outputError);
% Calculate hidden hidden error
hiddenHiddenError = (transpose(hIIhIIIWeights) *
    ↪ hiddenIIHiddenIIIError);
% Calculate hidden error, error = output_error *
    ↪ h_o_weights
hiddenError = (transpose(hhIIWeights) *
    ↪ hiddenHiddenError);

% Remove index of image already calculated during
    ↪ this epoch
trainPointer(randomIndex) = [];

% Calculate total costs weight using partial
    ↪ derivatives
dcostHOWeights = outputError .* y .* (1-y) *
    ↪ transpose(hiddenIIHiddenIIIOutput);
dcostHIIHIIIWeights = hiddenIIHiddenIIIError .*
    ↪ hiddenIIHiddenIIIOutput .*(1-
    ↪ hiddenIIHiddenIIIOutput) * transpose(
    ↪ hiddenHiddenIIOutput);
dcostHHIIWeights = hiddenHiddenError .*
    ↪ hiddenHiddenIIOutput .* (1-
    ↪ hiddenHiddenIIOutput) * transpose(hiddenOutput
    ↪ );
dcostIHWeights = hiddenError .* hiddenOutput .* (1-
    ↪ hiddenOutput) * transpose(x);
```

```
            % Calculate total cost of biases, this is just the
                ↪ gradient
            dcostOutputBias = outputError .* y .* (1-y);
            dcostHiddenIIIBias = hiddenIIHiddenIIIError .*
                ↪ hiddenIIHiddenIIIOutput .* (1-
                ↪ hiddenIIHiddenIIIOutput);
            dcostHiddenIIBias = hiddenHiddenError .*
                ↪ hiddenHiddenIIOutput .*(1-hiddenHiddenIIOutput
                ↪ );
            dcostHiddenBias = hiddenError .* hiddenOutput .* (1-
                ↪ hiddenOutput);

            % Update parameters
            hoWeights = hoWeights + (lr.*dcostHOWeights);
            hIIhIIIWeights  = hIIhIIIWeights + (lr.*
                ↪ dcostHIIHIIIWeights);
            hhIIWeights = hhIIWeights + (lr.*dcostHHIIWeights);
            ihWeights = ihWeights + (lr.*dcostIHWeights);

            outputBias = outputBias - (lr.*dcostOutputBias);
            hiddenIIIBias = hiddenIIIBias - (lr.*
                ↪ dcostHiddenIIIBias);
            hiddenIIBias = hiddenIIBias - (lr.*dcostHiddenIIBias
                ↪ );
            hiddenBias = hiddenBias - (lr.*dcostHiddenBias);
        end %Loop for random inputs to complete epoch

        costPlot(i,para) = mean(MSE)/nFilesTotal;
        plot(costPlot);
        xlabel('Epoch','FontSize',18)
        ylabel('Mean Squared Error','FontSize',18)
        title('All Topographic Data Training','FontSize',24)
        legend('Axial Curvature','Corneal Thickness',...
            'Elevation Front', 'Elevation Back')
        drawnow;
    end

    % Save weight and bias matrices in their relevant folders
    save(strcat(fileName, 'hoWeights.mat'), 'hoWeights')
    save(strcat(fileName, 'hIIhIIIWeights.mat'), 'hIIhIIIWeights
        ↪ ')
    save(strcat(fileName, 'hhIIWeights.mat'), 'hhIIWeights')
    save(strcat(fileName, 'ihWeights.mat'), 'ihWeights')

    save(strcat(fileName, 'outputBias.mat'), 'outputBias')
    save(strcat(fileName, 'hiddenIIIBias.mat'), 'hiddenIIIBias')
    save(strcat(fileName, 'hiddenIIBias.mat'), 'hiddenIIBias')
    save(strcat(fileName, 'hiddenBias.mat'), 'hiddenBias')
end % Loop for training each parameter
```

Appendix L

Integration and Final Model Training

This appendix shows the integration and final model program created for this project with MATLAB. This program integrates all ANNs and trains the final model. This program is discussed in Section 4.3.3.

Listing L.1: Integration and Final Model Training.

```
%% ENG49112 Research Project
% Final Model Training Post Topographic Training
% Author:       Sean Thomson
% Date:         22AUG19
% Version:      5
% Version Note: Change to make forward propagation results to
    ↪ speed up the process.
%               All the heavy computation now carried out at the
    ↪  start instead of on every iteration
%               Bias changed to zero

% Initialise
clc, clear, close all

% Folders with saved data post training. Ensure this folder is
    ↪ in current
% MATLAB directory.
foldersName = ["axialMatrices\", "thicknessMatrices\","
    ↪ elevationFrontMatrices\", "elevationBackMatrices\"];

% Naming convention of data post processing. Note must ensure
    ↪ this matrix
% matches foldersName matrix.
aFM = "axialMatrixFM";
```

```
tFM = 'thicknessMatrixFM';
efFM = 'elevationFrontMatrixFM';
ebFM = 'elevationBackMatrixFM';
dataMatrix = [aFM, tFM, efFM, ebFM];

% Weights and bias naming convention
weightsBiasMatrix = ['hoWeights.mat','hIIhIIIWeights.mat','
    ↪ hhIIWeights.mat','ihWeights.mat','outputBias.mat','
    ↪ hiddenIIIBias.mat','hiddenIIBias.mat','hiddenBias.mat'];

% Load matrix of training data post processing
for i = 1:length(foldersName)
    load(strcat(foldersName(i),dataMatrix(i)));
end

%% Create inputs for final model

% Reset desired result to a value it can never be
desiredFlag = [0;0];
xFM = gpuArray(zeros(length(weightsBiasMatrix)+2,size(
    ↪ axialMatrixFM,2)));

% Loop to get each result incrementally
for fwdprop = 1:size(axialMatrixFM,2)
    % Reset desired flag
    desiredResult = desiredFlag;

    % Loop to find each folder with saved weights and bias from
    % individual topographic training.
    for i = 1:length(foldersName)

        for j = 1:length(weightsBiasMatrix)
            load(strcat(foldersName(i),weightsBiasMatrix(j)));
        end

        % Find current input for topographic parameter
        currentDataMatrix = eval(dataMatrix(i));
        x = gpuArray(currentDataMatrix(2:end-2,fwdprop));

        % Determine desired result and check that all inputs are
            ↪  the same pause if there is an error with desired
            ↪ results for the 4 input pairs
        if desiredResult == desiredFlag
            desiredResult = [currentDataMatrix(end-1,fwdprop);
                ↪ currentDataMatrix(end,fwdprop)];

        elseif desiredResult ~= [currentDataMatrix(end-1,fwdprop
            ↪ );currentDataMatrix(end,fwdprop)]
            sprintf(['Something seriously went wrong and inputs
                ↪ to network do not match'])
            pause
        end
```

```
        % Determine forward propagation for each topographic
            ↪ data post training
        [~,~,~,y] = fwdPropagationSigmoid3(ihWeights,hhIIWeights
            ↪ ,hIIhIIIWeights,hoWeights,hiddenBias,hiddenIIBias,
            ↪ hiddenIIIBias,outputBias,x);

        % Switch statement to allocate outputs for the final
            ↪ neural network
        % these are input for the next network
        switch dataMatrix(i)
            case aFM
                axialY = y;
            case tFM
                thicknessY = y;
            case efFM
                elevationFrontY = y;
            case ebFM
                elevationBackY = y;
        end
    end
    xFM(:,fwdprop) = [axialY; thicknessY; elevationFrontY;
        ↪ elevationBackY; desiredResult];
end % Loop to create all inputs

%% Neural network for end of final model
% Variables
noInputFM = length(dataMatrix)*2;
noHiddenFM = round(100*noInputFM);
noHiddenIIFM = round(10*noHiddenFM);
noHiddenIIIFM = round(2/3*noHiddenIIFM);
noOutputFM = 2;
% Learning rate
lr = 0.0025;
% Number of epochs for training
trainAmount = 150;
% Preallocate for speed
costPlot = gpuArray(zeros(1,trainAmount));

% Ensure weights are comparataive to network. This uses Xavier
% Initialisation technique.
weightSetIFM = sqrt(2/(noInputFM+noHiddenFM));
weightSetHIIFM = sqrt(2/(noHiddenFM+noHiddenIIFM));
weightSetHIIIFM = sqrt(2/(noHiddenIIFM+noHiddenIIIFM));
weightSetOFM = sqrt(2/(noHiddenIIIFM+noOutputFM));

% Weights from input to hidden layer
ihWeightsFM = gpuArray((rand(noHiddenFM,noInputFM).*weightSetIFM
    ↪ ) - (weightSetIFM/2));
% Weights from hidden to hiddenII layer
hhIIWeightsFM = gpuArray((rand(noHiddenIIFM,noHiddenFM).*
    ↪ weightSetHIIFM) - (weightSetHIIFM/2));
```

```
% Weights from hiddenII to hiddenIII layer
hIIhIIIWeightsFM = gpuArray((rand(noHiddenIIIFM,noHiddenIIFM).*
    ↪ weightSetHIIIFM) - (weightSetHIIIFM/2));

% Weights from hidden layer to output
hoWeightsFM = gpuArray((rand(noOutputFM,noHiddenIIIFM).*
    ↪ weightSetOFM) - (weightSetOFM/2));
% Bias
hiddenBiasFM = gpuArray(zeros(noHiddenFM,1));
hiddenIIBiasFM = gpuArray(zeros(noHiddenIIFM,1));
hiddenIIIBiasFM = gpuArray(zeros(noHiddenIIIFM,1));
outputBiasFM = gpuArray(zeros(noOutputFM,1));

% Training loop
for epochNo = 1:trainAmount

    % Create pointer array for image selection
    trainPointer = 1:size(xFM,2);
    MSE = 0;

    % Loop to randomly go through all data to complete one epoch
    for randEpoch = 1:length(trainPointer)

        % Find random index then use index to point to next
            ↪ propagation
        randomIndex = randi(length(trainPointer),1);
        % Find next input and desired result
        xx = xFM(1:end-2,trainPointer(randomIndex));
        desiredResult = [xFM(end-1,trainPointer(randomIndex))
            ↪ ;...
                    xFM(end,trainPointer(randomIndex))];

        % Calculate forward propagation
        [hiddenOutputFM,hiddenHiddenIIOutputFM,
            ↪ hiddenIIHiddenIIIOutputFM,yFM] =
            ↪ fwdPropagationSigmoid3(ihWeightsFM,hhIIWeightsFM,
            ↪ hIIhIIIWeightsFM,hoWeightsFM,hiddenBiasFM,
            ↪ hiddenIIBiasFM,hiddenIIIBiasFM,outputBiasFM,xx);

        % Remove index of image already calculated during this
            ↪ epoch
        trainPointer(randomIndex) = [];

        % Calculate error, error = target - output
        outputErrorFM = desiredResult - yFM;
        MSE = MSE + (outputErrorFM.^2);

        % Calculate hidden III error to output, error =
            ↪ output_error
        % * h_o_weights
        hiddenIIHiddenIIIErrorFM = (transpose(hoWeightsFM) *
            ↪ outputErrorFM);
```

```
        % Calculate hidden hidden error
        hiddenHiddenErrorFM = (transpose(hIIhIIIWeightsFM) *
            ↪ hiddenIIHiddenIIIErrorFM);
        % Calculate hidden error, error = output_error *
            ↪ h_o_weights
        hiddenErrorFM = (transpose(hhIIWeightsFM) *
            ↪ hiddenHiddenErrorFM);

        % Calculate total costs weight using partial derivatives
        dcostHOWeightsFM = outputErrorFM .* yFM .* (1-yFM) *
            ↪ transpose(hiddenIIHiddenIIIOutputFM);
        dcostHIIHIIIWeightsFM = hiddenIIHiddenIIIErrorFM .*
            ↪ hiddenIIHiddenIIIOutputFM .* (1-
            ↪ hiddenIIHiddenIIIOutputFM) * transpose(
            ↪ hiddenHiddenIIOutputFM);
        dcostHHIIWeightsFM = hiddenHiddenErrorFM .*
            ↪ hiddenHiddenIIOutputFM .* (1-
            ↪ hiddenHiddenIIOutputFM) * transpose(hiddenOutputFM
            ↪ );
        dcostIHWeightsFM = hiddenErrorFM .* hiddenOutputFM .*(1-
            ↪ hiddenOutputFM) * transpose(xx);

        % Calculate total cost of biases, this is just the
            ↪ gradient
        dcostOutputBiasFM = outputErrorFM .* yFM .* (1-yFM);
        dcostHiddenIIIBiasFM = hiddenIIHiddenIIIErrorFM .*
            ↪ hiddenIIHiddenIIIOutputFM .*(1-
            ↪ hiddenIIHiddenIIIOutputFM);
        dcostHiddenIIBiasFM = hiddenHiddenErrorFM .*
            ↪ hiddenHiddenIIOutputFM .* (1-
            ↪ hiddenHiddenIIOutputFM);
        dcostHiddenBiasFM = hiddenErrorFM .* hiddenOutputFM .*
            ↪ (1-hiddenOutputFM);

        % Update parameters
        hoWeightsFM = hoWeightsFM + (lr.*dcostHOWeightsFM);
        hIIhIIIWeightsFM  = hIIhIIIWeightsFM + (lr.*
            ↪ dcostHIIHIIIWeightsFM);
        hhIIWeightsFM = hhIIWeightsFM + (lr.*dcostHHIIWeightsFM)
            ↪ ;
        ihWeightsFM = ihWeightsFM + (lr.*dcostIHWeightsFM);

        outputBiasFM = outputBiasFM - (lr.*dcostOutputBiasFM);
        hiddenIIIBiasFM = hiddenIIIBiasFM - (lr.*
            ↪ dcostHiddenIIIBiasFM);
        hiddenIIBiasFM = hiddenIIBiasFM - (lr.*
            ↪ dcostHiddenIIBiasFM);
        hiddenBiasFM = hiddenBiasFM - (lr.*dcostHiddenBiasFM);
%         disp(randEpoch)
    end % Loop to go through all images

    % Plot to follow progress
```

```
    costPlot(epochNo) = mean(MSE)/size(axialMatrixFM,2);
    plot(costPlot);
    xlabel('Epoch','FontSize',18)
    ylabel('Mean Squared Error','FontSize',18)
    title('Final Model Training','FontSize',24)
    drawnow;

end % Training loop

% Save weight and bias matrices in their relevant folders
save('finalModel\hoWeightsFM.mat', 'hoWeightsFM')
save('finalModel\hIIhIIIWeightsFM.mat', 'hIIhIIIWeightsFM')
save('finalModel\hhIIWeightsFM.mat', 'hhIIWeightsFM')
save('finalModel\ihWeightsFM.mat', 'ihWeightsFM')

save('finalModel\outputBiasFM.mat', 'outputBiasFM')
save('finalModel\hiddenIIIBiasFM.mat', 'hiddenIIIBiasFM')
save('finalModel\hiddenIIBiasFM.mat', 'hiddenIIBiasFM')
save('finalModel\hiddenBiasFM.mat', 'hiddenBiasFM')
```

Appendix M

Complete Model Test

This appendix shows the program used for testing the final model, and it is created with MATLAB. This complete model tests program takes new data and provides a prediction. The input is a Pentcam four maps refractive image, and the output is a prediction of the patient having KC or not. This program is discussed in Section 4.3.4.

Listing M.1: Complete Model Test.

```
%% ENG49112 Research Project
% Final Model Testing using independent data
% Author: Sean Thomson
% Date: 29AUG19
% Version: 6
% Version Note: This uses all previous testing to utilise
    ↪ optimum parameters and settings.
%               Changed data augmentation as not required.
%               Save processing for application.
%               This will test independent control data.
%               Uncomment as stated within to check KCN data.

% Initialise
clc, clear, close all

%% Variables
% Current images for testing. Change here to test 1 image at a
    ↪ time.
% controlTest = 1;
% KCTest = 1;

% Count variable
KCNCount = 0;
controlCount = 0;
unsureCount = 0;
```

```
% Dimensionality reduction from results. Note RGB follows 123
    ↪ convention
axialDim = 2;          %green
thicknessDim = 2;      %green
elevationFDim = 3;     %blue
elevationBDim = 2;     %green

% Data reduction and augmentation
noRotations = 4;
reduceFactor = .1;

% Folders with saved data post training. Ensure this folder is
    ↪ in current
% MATLAB directory.
foldersNameWB = ["axialMatrices\", "thicknessMatrices\","
    ↪ elevationFrontMatrices\", "elevationBackMatrices\"];

% Weights and bias naming convention
weightsBiasMatrix = ["hoWeights","hIIhIIIWeights","hhIIWeights
    ↪ ","ihWeights","outputBias","hiddenIIIBias","hiddenIIBias
    ↪ ","hiddenBias"];

% Folder and naming convention for final model results
folderFinal = "finalModel\";
weightsBiasFinal = ["hoWeightsFM","hIIhIIIWeightsFM","
    ↪ hhIIWeightsFM","ihWeightsFM","outputBiasFM","
    ↪ hiddenIIIBiasFM","hiddenIIBiasFM","hiddenBiasFM"];

% Tolerance to give a confident result
toleranceFinal = 60; %percentage

%% Pre-processing

% Folder names with test data
mainFolder = "Test_data";
subFolder = ["control_2", "kc_2"];
controlPath = strcat(mainFolder,'\',subFolder(1),'\');
KCPath = strcat(mainFolder,'\',subFolder(2),'\');

% Get a list of all images in control and KCN folder. Note the
    ↪ top folder must be in current MATLAB dir.
imageFilesControl = dir(strcat(controlPath,'*.jpg'));
imageFilesKC = dir(strcat(KCPath,'*.jpg'));

% Find number of files
nFilesControl = length(imageFilesControl);
nFilesKC = length(imageFilesKC);

% Change comments if want to test KCN folder
% Loop to test all KCN images
%for KCInt = 1:nFilesKC
% Loop to test all control images
```

```
for contInt = 1:nFilesControl

% Read the current image for testing. Change comments if want to
    ↪ test KCN
    currentFileName = imageFilesControl(contInt).name;
    fileName = strcat(controlPath, currentFileName);
    %currentFileName = imageFilesControl(KCInt).name;
    %fileName = strcat(KCPath, currentFileName);

    image = imread(fileName);

    % Separate, crop, and remove irrelevant information from
        ↪ current
    % test image
    [axial, thickness, elevFront, elevBack] = refractiveMapCrop(
        ↪ image);

    % Output after first pre-processing step
    figure(1)
    subplot(2,2,1), imshow(axial)
    subplot(2,2,2), imshow(thickness)
    subplot(2,2,3), imshow(elevFront)
    subplot(2,2,4), imshow(elevBack)
    sgtitle('Output after seperating image into individual
        ↪ topographic maps','FontSize',24)

    figure(3)
    imshow(axial)
    title('Axial / Sagittal Curvature (Front)')
    saveas(gcf,'1st top left','jpg')
    figure(4)
    imshow(thickness)
    title('Corneal Thickness')
    saveas(gcf,'1st bottom left','jpg')
    figure(5)
    imshow(elevFront)
    title('Elevation (Front)')
    saveas(gcf,'1st top right','jpg')
    figure(6)
    imshow(elevBack)
    title('Elevation (Back)')
    saveas(gcf,'1st bottom right','jpg')

    % Dimensionality reduction
    axialD = (axial(:,:,axialDim));
    thicknessD = (thickness(:,:,thicknessDim));
    elevFrontD = (elevFront(:,:,elevationFDim));
    elevBackD = (elevBack(:,:,elevationBDim));

    % Output after dimensionality reduction pre-processing step
    figure(2)
    subplot(2,2,1), imshow(axialD)
```

```
subplot(2,2,2), imshow(thicknessD)
subplot(2,2,3), imshow(elevFrontD)
subplot(2,2,4), imshow(elevBackD)
sgtitle('Dimensionality Reduction','FontSize',24)

figure(6)
imshow(axialD)
title('Axial / Sagittal Curvature (Front)')
xlabel('Green Channel')
saveas(gcf,'2nd_top_left','jpg')
figure(7)
imshow(thicknessD)
title('Corneal Thickness')
xlabel('Green Channel')
saveas(gcf,'2nd_bottom_left','jpg')
figure(8)
imshow(elevFrontD)
title('Elevation (Front)')
xlabel('Blue Channel')
saveas(gcf,'2nd_top_right','jpg')
figure(9)
imshow(elevBackD)
title('Elevation (Back)')
xlabel('Green Channel')
saveas(gcf,'2nd_bottom_right','jpg')

% Data size reduction
axialReduced = imresize(axialD,reduceFactor);
thicknessReduced = imresize(thicknessD,reduceFactor);
elevFrontReduced = imresize(elevFrontD,reduceFactor);
elevBackReduced = imresize(elevBackD,reduceFactor);

% Data augmentation if deemed necessary
axialReduced = imrotate(axialReduced,0);
thicknessReduced = imrotate(thicknessReduced,0);
elevFrontReduced = imrotate(elevFrontReduced,0);
elevBackReduced = imrotate(elevBackReduced,0);

% Normalise data and increase limits 2 to -2. Training was
    ↪ done with this parameter to improve sigmoid activation
    ↪  function results.
axialReduced = (im2single(axialReduced).*4) - 2;
thicknessReduced = (im2single(thicknessReduced).*4) - 2;
elevFrontReduced = (im2single(elevFrontReduced).*4) - 2;
elevBackReduced = (im2single(elevBackReduced).*4) - 2;

% Create single column for neural network input
axialReduced = axialReduced(:);
thicknessReduced = thicknessReduced(:);
elevFrontReduced = elevFrontReduced(:);
elevBackReduced = elevBackReduced(:);
```

```
%% Forward propagation for each topographic map

% Loop to find folder with each individual weights and bias
    ↪ saved
for i = 1:length(foldersNameWB)

    % Loop to load each weight and bias matrix
    for j = 1:length(weightsBiasMatrix)
        load(strcat(foldersNameWB(i),weightsBiasMatrix(j),'.
            ↪ mat'));
    end

    % Switch statement to find which topographic map forward
        ↪  propagation to calculate. This calls the sigmoid
        ↪ propagation function.
    switch foldersNameWB(i)
        case 'axialMatrices\'
            [~,~,~,yAxial] = fwdPropagationSigmoid3(
                ↪ ihWeights,hhIIWeights,hIIhIIIWeights,
                ↪ hoWeights,hiddenBias,hiddenIIBias,
                ↪ hiddenIIIBias,outputBias,axialReduced);
        case 'thicknessMatrices\'
            [~,~,~,yThickness] = fwdPropagationSigmoid3(
                ↪ ihWeights,hhIIWeights,hIIhIIIWeights,
                ↪ hoWeights,hiddenBias,hiddenIIBias,
                ↪ hiddenIIIBias,outputBias,thicknessReduced)
                ↪ ;
        case 'elevationFrontMatrices\'
            [~,~,~,yElevFront] = fwdPropagationSigmoid3(
                ↪ ihWeights,hhIIWeights,hIIhIIIWeights,
                ↪ hoWeights,hiddenBias,hiddenIIBias,
                ↪ hiddenIIIBias,outputBias,elevFrontReduced)
                ↪ ;
        case 'elevationBackMatrices\'
            [~,~,~,yElevBack] = fwdPropagationSigmoid3(
                ↪ ihWeights,hhIIWeights,hIIhIIIWeights,
                ↪ hoWeights,hiddenBias,hiddenIIBias,
                ↪ hiddenIIIBias,outputBias,elevBackReduced);
    end
end

%% Forward propagation for the final network

% Combine each result for input to final network
yCombined = [yAxial; yThickness; yElevFront; yElevBack];

% Loop to load final model weight and biases matrices
for j = 1:length(weightsBiasFinal)
    load(strcat(folderFinal,weightsBiasFinal(j),'.mat'));
end

% Calculate forward propagation
```

```
    [hiddenOutputFM,hiddenHiddenIIOutputFM,
        ↪ hiddenIIHiddenIIIOutputFM,yFM] =fwdPropagationSigmoid3
        ↪ (ihWeightsFM,hhIIWeightsFM,hIIhIIIWeightsFM,
        ↪ hoWeightsFM,hiddenBiasFM,hiddenIIBiasFM,
        ↪ hiddenIIIBiasFM,outputBiasFM,yCombined);

    % Convert output to percentage
    yFM = yFM.*100;
    fprintf('File name tested: \n %s \n',fileName)
    if max(yFM) < toleranceFinal
        fprintf('The model is not more than %3.2f%% certain.\n \
            ↪ n',toleranceFinal)
        unsureCount = unsureCount + 1;
    elseif yFM(1) < yFM(end)
        fprintf('Control\n')
        fprintf('It predicts this with %3.2f%% certainty.\n \n',
            ↪ yFM(end))
        controlCount = controlCount + 1;
    else
        fprintf('Keratoconus\n')
        fprintf('It predicts this with %3.2f%% certainty.\n \n',
            ↪ yFM(1))
        KCNCount = KCNCount + 1;
    end


end

fprintf('It predicts KCN %3.2f times.\n',KCNCount)
fprintf('It predicts Control %3.2f times.\n',controlCount)
fprintf('It is unsure %3.2f times.\n',unsureCount)
```

Appendix N

Mobile App Main Screen Source Code

This appendix shows the main screen source code for the mobile app. This source code is written in the programming language Kotlin, and it was developed in Android Studio. This code will read off its corresponding XML layout structure to populate the images for display. This is further discussed in Section 6.4.

Listing N.1: Mobile App Main Screen Source Code.

```
%package com.aeye

import com.aeye.db.dbConstants
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.View
import kotlinx.android.synthetic.main.activity_main.*
import org.json.JSONArray
import org.json.JSONException
import org.json.JSONObject
import android.R.attr.data

import com.aeye.model.ImageModel
import android.util.Log
import java.util.ArrayList
import android.widget.AdapterView.OnItemClickListener
import android.content.Intent
import android.widget.*
import android.widget.EditText
import java.io.Serializable
```

```
const val PAGE_ONE = "com.aeye.DisplayFirstImagesActivity"

var strJson = dbConstants.JSON_STRING
var jsonRootObject: JSONObject = JSONObject(strJson)
var jsonArray: JSONArray = jsonRootObject.optJSONArray("Images")

class MainActivity : AppCompatActivity(), AdapterView.OnItemClickListener
    ↪ {
    private var lv: ListView? = null
    private var customeAdapter: CustomAdapter? = null
    private var imageModelArrayList: ArrayList<ImageModel>? = null


    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        lv = findViewById(R.id.listView) as ListView

        imageModelArrayList = populateList()
        customeAdapter = CustomAdapter(this, imageModelArrayList!!)
        lv!!.adapter = customeAdapter

        lv!!.setOnItemClickListener(this)
    }



    private fun populateList(): ArrayList<ImageModel> {
        val list = ArrayList<ImageModel>()

        var image = ""

        for (item in 0..(jsonArray.length() - 1)) {
            val imageModel = ImageModel()
            var jsonObject: JSONObject = jsonArray.getJSONObject(item)
            val id = Integer.parseInt(jsonObject.optString("id").toString
                ↪ ())
            imageModel.setNames(jsonObject.optString("imageName").toString
                ↪ ())

            // sets the model to have the images for the first page
            val drawableIdStringFirstTopLeft = jsonObject.optString("
                ↪ firstTopLeft").toString()
            val drawableIdFirstTopLeft = getResources().getIdentifier(
                ↪ drawableIdStringFirstTopLeft,"drawable", packageName);
            imageModel.setfirstTopLeftLocation(drawableIdFirstTopLeft)

            val drawableIdStringFirstTopRight = jsonObject.optString("
                ↪ firstTopRight").toString()
            val drawableIdFirstTopRight = getResources().getIdentifier(
                ↪ drawableIdStringFirstTopRight,"drawable", packageName);
```

```
imageModel.setfirstTopRightLocation(drawableIdFirstTopRight)

val drawableIdStringFirstBottomLeft = jsonObject.optString("
    ↪ firstBottomLeft").toString()
val drawableIdFirstBottomLeft = getResources().getIdentifier(
    ↪ drawableIdStringFirstBottomLeft,"drawable", packageName)
    ↪ ;
imageModel.setfirstBottomLeftLocation(
    ↪ drawableIdFirstBottomLeft)

val drawableIdStringFirstBottomRight = jsonObject.optString("
    ↪ firstBottomRight").toString()
val drawableIdFirstBottomRight = getResources().getIdentifier(
    ↪ drawableIdStringFirstBottomRight,"drawable", packageName
    ↪ );
imageModel.setfirstBottomRightLocation(
    ↪ drawableIdFirstBottomRight)

// sets the model to have the images for the second page

val drawableIdStringSecondTopLeft = jsonObject.optString("
    ↪ secondTopLeft").toString()
val drawableIdSecondTopLeft = getResources().getIdentifier(
    ↪ drawableIdStringSecondTopLeft,"drawable", packageName);
imageModel.setSecondTopLeftLocation(drawableIdSecondTopLeft)

val drawableIdStringSecondTopRight = jsonObject.optString("
    ↪ secondTopRight").toString()
val drawableIdSecondTopRight = getResources().getIdentifier(
    ↪ drawableIdStringSecondTopRight,"drawable", packageName);
imageModel.setSecondTopRightLocation(drawableIdSecondTopRight)

val drawableIdStringSecondBottomLeft = jsonObject.optString("
    ↪ secondBottomLeft").toString()
val drawableIdSecondBottomLeft = getResources().getIdentifier(
    ↪ drawableIdStringSecondBottomLeft,"drawable", packageName
    ↪ );
imageModel.setSecondBottomLeftLocation(
    ↪ drawableIdSecondBottomLeft)

val drawableIdStringSecondBottomRight = jsonObject.optString("
    ↪ secondBottomRight").toString()
val drawableIdSecondBottomRight = getResources().getIdentifier
    ↪ (drawableIdStringSecondBottomRight,"drawable",
    ↪ packageName);
imageModel.setSecondBottomRightLocation(
    ↪ drawableIdSecondBottomRight)

// sets the image data
imageModel.setModelPredictions(jsonObject.optString("
    ↪ modelPrediction").toString())
```

```
            imageModel.setModelCertainties(jsonObject.optString("
                ↪ modelCertainty").toString())
            imageModel.setActualDiagnosises(jsonObject.optString("
                ↪ actualDiagnosis").toString())



            // sets the images in the list on first page
            val drawableIdString = jsonObject.optString("location").
                ↪ toString()
            val drawableId = getResources().getIdentifier(drawableIdString
                ↪ ,"drawable", packageName);
            imageModel.setImageDrawables(drawableId)
            list.add(imageModel)
        }

        return list
    }

    override fun onItemClick(p0: AdapterView<*>?, view: View?, position:
        ↪ Int, p3: Long) {
        val imageModel = lv!!.getItemAtPosition(position) as ImageModel
        Toast.makeText(applicationContext, "Superfluous Information
            ↪ Removed", Toast.LENGTH_LONG).show()
        val intent = Intent(this, DisplayFirstImagesActivity::class.java).
            ↪ apply {
            putExtra(PAGE_ONE, imageModel)
        }

        startActivity(intent)
    }
}
```