QATAR UNIVERSITY

COLLEGE OF ENGINEERING

NEUROPATHY CLASSIFICATION OF CORNEAL NERVE IMAGES USING ARTIFICIAL INTELLIGENCE

BY

TOOBA SALAHUDDIN

A Thesis Submitted to
the Faculty of the College of Engineering
in Partial Fulfillment of the Requirements for the Degree of
Masters of Science in Computing

June 2019

COMMITTEE PAGE

The members of the Committee approve the Thesis of
Tooba Salahuddin defended on 16/04/2019.

Dr. Sumaya AlMaadeed
Thesis/Dissertation Supervisor

Dr. Uvais Ahmed Qidwai
Co-supervisor

Dr. Abbes Amira
Committee Member

Dr. Siti Anom
External Examiner

Approved:

Abdel Magid Hamouda , Dean, College of Engineering

ABSTRACT

SALAHUDDIN, TOOBA, Masters : June : 2019, Masters of Science in Computing

Title: Neuropathy Classification of Corneal Nerve Images Using Artificial Intelligence

Supervisor of Thesis: Sumaya AlMaadeed.

Nerve variations in the human cornea have been associated with alterations in the neuropathy state of a patient suffering from chronic diseases. For some diseases, such as diabetes, detection of neuropathy prior to visible symptoms is important, whereas for others, such as multiple sclerosis, early prediction of disease worsening is crucial. As current methods fail to provide early diagnosis of neuropathy, in vivo corneal confocal microscopy enables very early insight into the nerve damage by illuminating and magnifying the human cornea. This non-invasive method captures a sequence of images from the corneal sub-basal nerve plexus. Current practices of manual nerve tracing and classification impede the advancement of medical research in this domain. Since corneal nerve analysis for neuropathy is in its initial stages, there is a dire need for process automation.

To address this limitation, we seek to automate the two stages of this process: nerve segmentation and neuropathy classification of images. For nerve segmentation, we compare the performance of two existing solutions on multiple datasets to select the appropriate method and proceed to the classification stage. Consequently, we approach neuropathy classification of the images through artificial intelligence using Adaptive Neuro-Fuzzy Inference System, Support Vector Machines, Naïve Bayes and k-nearest neighbors. We further compare the performance of machine learning classifiers with deep learning. We ascertained that nerve segmentation using convolutional neural networks provided a significant improvement in sensitivity and false negative rate by at least 5% over the state-of-the-art software. For classification, ANFIS yielded the best classification accuracy of 93.7% compared to other classifiers. Furthermore, for this problem, machine learning approaches performed better in terms of classification accuracy than deep learning.

DEDICATION

For my dad, I wish you were here to see this.

I’m sure I would have made you proud.

ACKNOWLEDGMENTS

First and foremost, I would like to thank my Creator and Sustainer, Allah SWT, the One who blessed me with the capability to work for this thesis and opened ways for me which I had never imagined. Indeed, my being and efforts alone hold no significance.

I am very grateful to my family, for their unconditional love for me, for their constant moral and emotional support, and for their trust in me. I would not have been able to do this without you all.

My deepest gratitude to my unofficial supervisor, Dr. Ioannis Petropoulos, whose valuable feedback, constant guidance, ideas, patience and efforts, despite his tough schedule, are a major contribution to this success. Special thanks to the research team at Hamad Medical Centre and Weill Cornell Medicine, Qatar, for providing all the necessary data required for this study.

I feel heavily indebted to Dr Uvais Qidwai for his unwavering patience, meticulous supervision, constant encouragement, thorough review and timely support in all phases of this thesis. Indeed, I am very fortunate to have him as my mentor and my journey in research is inspired from his dedication and wisdom. Special thanks to Dr. Sumaya for her cooperation and supervision in this research.

Last, but by no means least, I feel at a loss of words to thank Rahma and Sara, who were always there to listen to me and helped me stand up when I felt like losing; you have no idea how much it meant to me. I would not forget Reem and Zeineb, who bring me happiness and inspiration; as well as Hanadi, Nandhini, Afnan, Elaf and others who have never let me down.

TABLE OF CONTENTS

DEDICATION v

ACKNOWLEDGMENTS vi

LIST OF TABLES xi

LIST OF FIGURES xii

LIST OF ABBREVIATIONS xiii

CHAPTER 1: INTRODUCTION 1

- 1.1. Neuropathy 1
- 1.2. Current Methods for Detection of Neuropathy 2
- 1.3. Corneal Confocal Microscopy 3
- 1.4. Limitations 5
- 1.5. Problem Statement 6
- 1.6. Research Questions 6
- 1.7. Research Objectives 6
- 1.8. Solution Overview 7

CHAPTER 2: BACKGROUND AND RELATED WORK 9

- 2.1. Background 9
  - 2.1.1. Adaptive Nero-Fuzzy Inference System 9
  - 2.1.2. Support Vector Machines 10
  - 2.1.3. Naïve Bayes 11

2.1.4. K-Nearest Neighbor 12

2.1.5. U-Net 13

2.1.6. GoogleNet 15

2.2. Image Processing Techniques for Segmentation of Curvilinear Structures .. 17

2.2.1. Region of Interest Expansion 17

2.2.2. Gabor Wavelets 19

2.2.3. Morphological Operations 19

2.2.4. Pixel Classification using Machine Learning 20

2.2.5. Hessian Matrix-based Approach 21

2.2.6. Pixel Classification using Convolutional Neural Networks 22

2.3. Classification Techniques for Retinal and Corneal Images 23

2.3.1. Transfer Learning and Parameter Tuning 24

2.3.2. Combination of Pretrained Models and Ensembles 26

2.3.3. Modification of Available Neural Networks 27

2.3.4. Custom Designed Convolutional Neural Networks 28

2.3.5. Machine Learning 29

CHAPTER 3: METHODOLOGY 35

3.1. Image Acquisition 35

3.2. Ground Truth Annotation 38

3.3. Nerve Segmentation Using Convolutional Neural Networks 39

3.4. Feature Extraction ... 39

3.5. Neuropathy Classification using Machine Learning ... 40

3.6. Hierarchical ANFIS Classification Model ... 40

3.7. Neuropathy Classification using Transfer Learning ... 43

3.8. Experimental Procedures for Segmentation ... 43

  3.8.1. Preprocessing and Selection of Training Images ... 44

  3.8.2. Training ... 45

  3.8.3. Postprocessing ... 48

3.9. Experimental Procedures for Classification ... 49

  3.9.1. Preprocessing ... 49

  3.9.2. Experimental Setup ... 49

3.10. Evaluation Metrics ... 50

  3.10.1. Sensitivity ... 50

  3.10.2. Specificity ... 50

  3.10.3. Precision ... 50

  3.10.4. Accuracy ... 51

  3.10.5. False Negative Rate ... 51

  3.10.6. False Positive Rate ... 51

  3.10.7. False Detection Rate ... 51

  3.10.8. Macro-F1 ... 51

CHAPTER 4: RESULTS AND DISCUSSION 53

4.1. Nerve Segmentation 53

4.2. Neuropathy Classification 55

4.2.1. ANFIS 55

4.2.2. SVM 57

4.2.3. KNN 59

4.2.4. Naïve Bayes 61

4.2.5. GoogleNet 62

4.2.6. Machine Learning and Deep Learning Results Comparison 63

4.2.7. Training and Prediction Time Analysis 65

4.2.8. Model Size Analysis 65

4.2.9. Statistical Significance 66

CHAPTER 5: CONCLUSION 68

5.1. Future Research Directions 69

5.2. Related Publications 70

REFERENCES 72

LIST OF TABLES

Table 1. Summary of Segmentation Techniques for Curvilinear Structures 30

Table 2. Summary of Classification Techniques for Curvilinear *Structures* 33

Table 3. Subject Details for Each Dataset..37

Table 4. Actual Classification of Images in Each Dataset 38

Table 5. Model Summary of the U-Net Based CNN...46

Table 6. Segmentation Results 54

Table 7. ANFIS Classification Results..56

Table 8. SVM Classification Results..57

Table 9. SVM Classification Results for the Polynomial Kernel.................................58

Table 10. KNN Classification Results..59

Table 11. Naïve Bayes Classification Results..61

Table 12. All Classification Results (Test Accuracy)...63

*Table 13. Training and Prediction Times** 65

Table 14. Model Sizes 66

Table 15. Kappa Statistics and p-values...67

LIST OF FIGURES

Figure 1. Current methods for detecting neuropathy 3

Figure 2. Corneal Confocal Microscope 4

Figure 3. The 5 layers of the cornea 5

Figure 4. First approach 7

Figure 5. Second approach 8

Figure 6. Architecture of ANFIS 10

Figure 7. KNN with $k = 3$ 13

Figure 8. The U-Net architecture 14

*Figure 9. GoogLeNet architecture** 16

Figure 10. A magnified representation of the subbasal nerve plexus of the human cornea. 36

Figure 11. Image slicing example 40

Figure 12 Hierarchical ANFIS classification model 42

Figure 13. Hierarchical ANFIS prediction workflow 43

Figure 14. Images with artefacts selected for training 45

Figure 15. The postprocessing pipeline 48

Figure 16. Increase in true positives using U-net 54

Figure 17. Reducing false positives through U-net 55

Figure 18. Effect of the value of k on accuracy 60

Figure 19. GoogleNet tuning graphs. (a) tuning using the RMSProp solver, (b) tuning using the SGDM solver. 62

Figure 20. Classification results from all techniques 64

LIST OF ABBREVIATIONS

ANFIS	Adaptive Neuro-Fuzzy Inference System
CCM	Corneal Confocal Microscope
CNFL	Corneal Nerve Fiber Length
CNN	Convolutional Neural Network
FDR	False Detection Rate
FIS	Fuzzy Inference System
FNR	False Negative Rate
FPR	False Positive Rate
GAN	Generative Adversarial Network
HED	Holistically Nested Edge Detection
ISNFL	Intra-segment Nerve Fiber Length
KNN	K-Nearest Neighbour
LMS	Least Means Square
NB	Naïve Bayes
NCS	Nerve Conduction Studies
NFL	Nerve Fiber Length
NN	Neural Network
PCA	Principal Component Analysis
QST	Quantitative Sensory Testing
RBF	Radial Basis Function
ReLU	Rectified Linear Unit
RF	Random Forest

ROI	Region of Interest
SGDM	Stochastic Gradient Descent
SVM	Support Vector Machine

CHAPTER 1: INTRODUCTION

Neuropathy is a long-term complication of various chronic diseases such as diabetes, multiple sclerosis, stroke, Parkinson’s disease, human immunodeficiency virus etc. Early detection of neuropathy is crucial for prolonging the quality of life in diabetic patients. In fact, current state-of-the-art methods fail to diagnose neuropathy at an early stage. Fortunately, imaging of corneal nerves through corneal confocal microscopy provides a way for analyzing subclinical neuropathy. However, the establishment of this method as a biomarker for neuropathy detection is hindered by tedious manual analysis.

1.1. Neuropathy

Neuropathy (axonal degeneration) is a painful neurological disorder distinguished by nerve malfunction. It may be a result of inflammatory diseases, surgical interventions, genetic disorders or infections, amongst other possible causes. Peripheral neuropathy is distinguished by numbness in the limbs and is the most prevalent complication of diabetes. Other visible effects of neuropathy include foot ulceration [1]. One of the earliest hidden symptoms of neuropathy is small fiber nerve damage and is apparent in a very early stage prior to the occurrence of visible symptoms [2]. On the other hand, visible symptoms of neuropathy occur only when the damage has reached the long nerve fibers. Therefore, accurate and timely diagnosis of neuropathy is necessary for prognosis, early recognition of subclinical neuropathy, monitoring disease maturity, classifying disease severity and suggesting relevant therapy plans [3].

1.2. Current Methods for Detection of Neuropathy

Current gold standards for evaluating nerve damage are through the established methods of quantitative sensory testing (QST), electrophysiology, nerve conduction studies (NCS) and skin biopsy [4]. QST determines the pain sensation through pinpricks, and exposure to warm and cold temperatures. Since the perception of pain varies from person to person, this method is prone to subjectivity in terms of quantifying neuropathy. Moreover, QST is limited in its usage as it is more applicable for detecting long nerve fiber damage [4] since loss of pain sensation is apparent only after long nerve fiber loss. Autonomic sensory testing is another method to assess autonomic neuropathy, but it is also reported to be inconvenient and inaccurate [5]. It consists of a tilt-table test, whereby the subject is made to lie on a flat table and then the table is tilted. The tilting of the table is supposed to induce fear of falling in the person which should increase his heart rate and breathing rate, implying the presence of autonomic senses. If the autonomic senses of the person have stopped working, then the person will not feel anything, and his breathing rate and pulse will not be affected. However, a person can practice staying calm in such a situation, which will also not affect his heart and breathing rate, thus making the technique unsuitable for accurate results. Moreover, the method will only detect long nerve fiber damage since loss of autonomic senses is only apparent when damage has reached long nerve fibers.

Nerve conduction studies use the passing of electric signals from one point to another in the nerve to be tested. If the signal successfully reaches the second point in the nerve it implies that the nerve is not damaged. This method requires expert skill for valid analysis and is only able to detect long nerve damage since small nerves are microscopic and cannot be seen by the naked eye to test the presence of electrical signals at a point. While skin biopsy can detect intra epidermal small nerve fiber loss and provides an objective assessment, it is an invasive and costly technique and cannot be conducted repeatedly [4], [6]. Moreover, it is time consuming and requires a person with considerable skill and expertise to conduct the test [6]. Fig. 1 summarizes the current methods for detecting neuropathy.

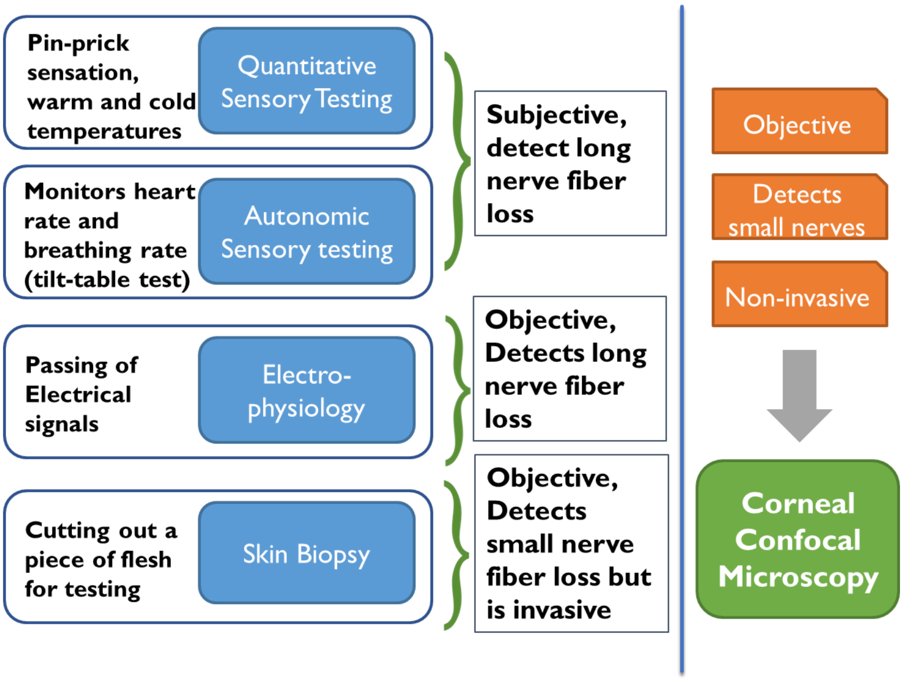


Figure 1. Current methods for detecting neuropathy

1.3. Corneal Confocal Microscopy

Recently, in vivo corneal confocal microscopy (CCM) has appeared as an indispensable tool in clinical trials for preliminary diagnosis of symptomatic neuropathy. It also provides the opportunity to monitor disease progression and evaluate the effectiveness of newly proposed treatments [7]. Small fiber nerves are present in the human cornea and corneal transparency allows for an in-depth observation through a microscope. Corneal nerve mages can be obtained using a corneal confocal microscope, shown in Fig. 2, for corneal morphological analysis in about 10 minutes in a non-invasive manner [5]. CCM images reveal a detailed and magnified structure of layers of the densely innervated cornea of the human eye. These images are captured using varying depths from the corneal surface. The images that display the subbasal nerve plexus are captured at a focal depth of about 40μm from the corneal surface [8]. This section lies between the epithelium and Bowman's layer. Fig. 3 shows the five layers of the cornea.

Figure 2. Corneal Confocal Microscope. (The Rostock corneal module is pointed to by the arrow)

The utility of CCM images in clinical practice is multifold. Minor changes in the corneal nerve architecture indicating an initial stage of neuropathy can be discovered by examining these images. This allows for an early diagnosis of neuropathy in patients suffering from certain chronic diseases, whereas visible symptoms become apparent much later. Studies have shown evidences of small nerve fiber loss in the cornea of patients identified with diabetes [5], [9], Parkinson's disease [10], [11], multiple sclerosis [12], stroke [9], HIV [13] etc. CCM images can also be used to rapidly quantify the severity level of different types of neuropathies. Moreover, they can also be helpful in defining new treatments by direct observation of their effect on the nervous system.

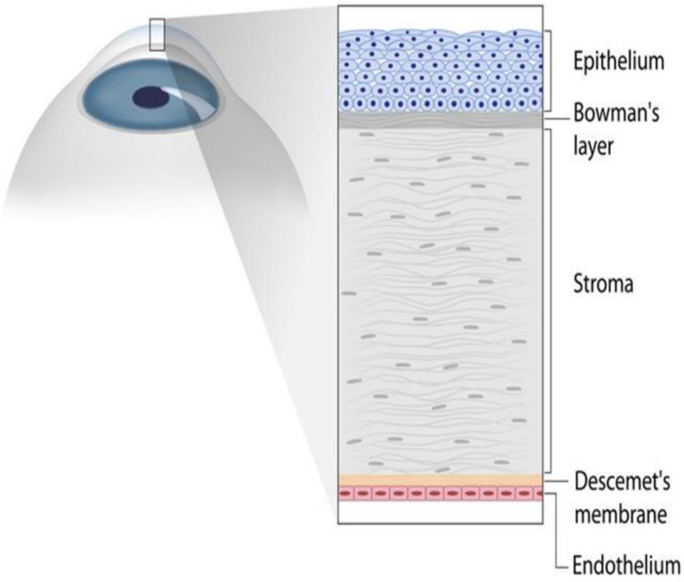


Figure 3. The 5 layers of the cornea

1.4. Limitations

CCM image analysis, being a promising tool, poses certain limitations in terms of its clinical applications and adaptations. In a recent article [14], Lagali strongly emphasizes on the need for the automation of corneal image segmentation, tracing, quantification and classification using artificial intelligence techniques. He also outlines the limitations as follows. Current limitations of CCM include (a) the tedious process of manual nerve tracing by clinicians for nerve parameter quantification and (b) manual classification of images to define the extent of nerve deficit. Rapid, accurate, objective and automated quantification of CCM nerve images through the application of image processing techniques is quiet a challenging task and currently in its early stages. Commonly used tools for CCM parameter quantification include CCMetrics, ACCMetrics and ImageJ. An obvious limitation of the former software is that they require manual tracing of nerves for measuring the parameters. ACCmetrics (MA

Dabbah, Imaging Science and Biomedical Engineering, Manchester, UK) automatically traces the nerves but is claimed by the doctors as inaccurate [15]–[17]. The latter (ImageJ) is a software for neurite tracing adapted for corneal nerve tracing (available at http://www.imagescience.org/meijering/software/neuronj/). This software requires human aid in tracing the nerves and then calculates the required parameters. Furthermore, most of the time, the problem of detecting nerve damage in CCM images remains a manual practice by visual observation.

1.5. Problem Statement

Based on the above limitations, the need for a complete solution to detect neuropathy becomes inevitable. An ideal solution would automatically and intelligently classify raw CCM images based on their neuropathy severity. This may require the process of nerve segmentation as well. Therefore, the problem statement for this thesis is the development of an artificial intelligence model to predict neuropathy in CCM images.

1.6. Research Questions

In this thesis, we attempt to answer the following research questions:

RQ1: Do convolutional neural networks (CNNs) provide a better automated nerve segmentation solution as compared to the state-of-the-art ACCMetrics?

RQ2: What kind of features define neuropathy in CCM images?

RQ3: What kind of machine learning approach would best classify CCM images?

RQ4: For this problem, is transfer learning using pre-trained CNNs a better approach as compared to machine learning?

1.7. Research Objectives

The research objectives for this thesis are as follows:

- To automate the process of neuropathy detection in corneal images
- To determine whether CNNs would provide an improved segmentation of

corneal nerve images when compared to existing solutions

- To explore the potential of machine learning algorithms for neuropathy classification of corneal images
- To investigate whether machine learning would be better suited for neuropathy classification of corneal images as compared to deep learning

1.8. Solution Overview

We attempt to tackle the problem of neuropathy classification of corneal images with the aim of providing an automated solution for the concerned medical experts. For nerve segmentation, we compare two state-of-the-art approaches and determine which of them is more suited for the task. For classification, we solve the problem using two approaches. The first approach is to extract features from segmented images and then use the feature set to classify the images using machine learning techniques. Fig. 4 illustrates the process of the first approach. We explore the potential of four machine learning classifiers for this purpose. The second approach is to classify raw images using deep learning, more specifically, a pretrained CNN. In this approach, the nerve segmentation step is eliminated. Fig. 5 illustrates the process of the second approach. In the discussion, we compare the results obtained using the two approaches.

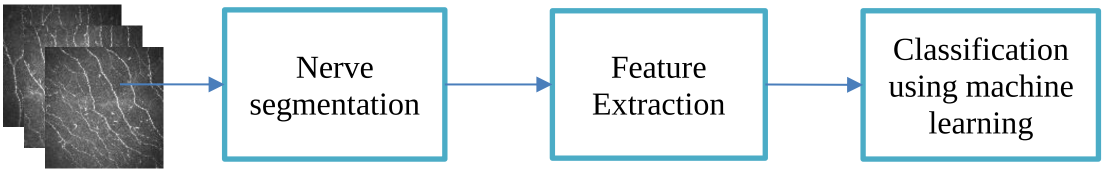


Figure 4. First approach

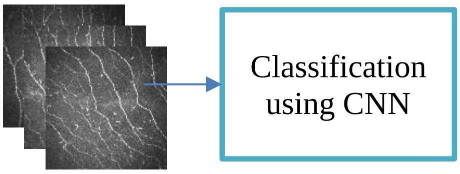


Figure 5. Second approach

The following chapter gives a background on the machine learning algorithms used in this study. Chapter 3 describes the methodology and experimental procedures for nerve segmentation and neuropathy classification. This is followed by results and discussion in Chapter 4. Finally, Chapter 5 concludes the research and provides some future directions for improvement and continuation of the study.

CHAPTER 2: BACKGROUND AND RELATED WORK

2.1. Background

This section covers some background information on the four machine learning techniques used in this research. An overview of the CNN used for image segmentation is also presented. Further, the pre-trained deep learning model used for the second approach to the solution is briefly explained.

2.1.1. Adaptive Nero-Fuzzy Inference System

Neuro-fuzzy systems refer to a subdivision of soft computing which synergizes the strength of artificial neural networks and fuzzy logic. Consequently, a robust framework is formed which is effectively used for solving machine learning problems. A popular implementation of Sugeno-based neuro-fuzzy systems is adaptive neuro-fuzzy inference systems (ANFIS) [18].

The ANFIS architecture comprises of five layers. In the first layer, the membership functions specify the membership degree of each input variable. These membership functions are formulated during the training phase. Using these membership functions, ANFIS creates a fuzzy inference system (FIS) which map the inputs to their corresponding outputs. The inferences from the rule base are used in the second and fourth layer to adjust the firing strength of each rule. The fourth layer generates the outputs using a linear polynomial equation. The last layer concatenates all outputs into a single output.

A two-pass learning algorithm is implemented during the learning stage [19]. The forward pass consists of updating the parameters using least squares estimation to produce the output. During the backward pass, error is computed across all layers and parameter values are updated accordingly using gradient descent algorithm.

The ANFIS network builds a FIS from the three input features, mapping them to the output using the membership functions. The membership functions use the hybrid learning technique for parameter tuning. Hence, the FIS is trained on the randomly selected training data. The architecture of the ANFIS network is displayed in Fig. 6. The figure shows only four layers because the second and third layers are displayed as one, namely the rule layer.

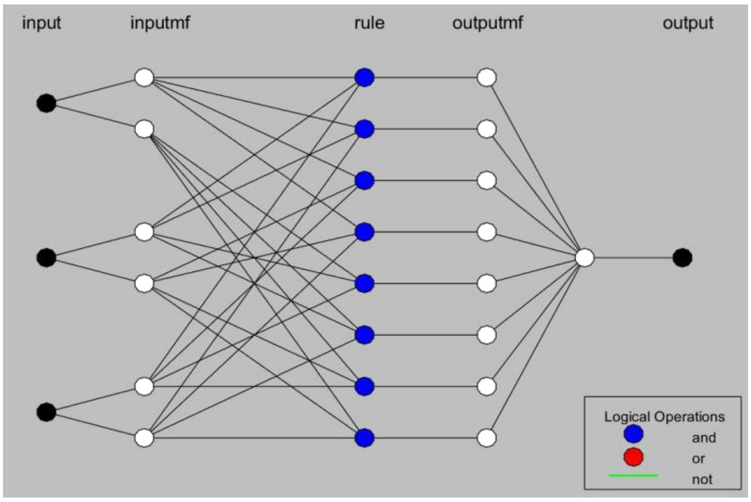


Figure 6. Architecture of ANFIS

2.1.2. Support Vector Machines

SVMs were originally introduced by Cortes & Vapnik [20] in 1995. They are known as universal learners because they usually perform well in most classification problems. SVM aims to create an optimal hyperplane with maximum margin, that separates the two classes of data. The points present nearest to the hyperplane are termed as support vectors, and they determine the position of the hyperplane. Consider a set of n training samples, $x_1, \dots, x_n$, each having a label from a set or r labels, $Y = y_1, \dots, y_r$. The SVM classifier creates a classifier of the form:

$$y(x) = sign[\sum_{i=1}^{n} \alpha_i y_i \psi(x, x_i) + b], \tag{1}$$

Where α_i belongs to a set of real constants, b is the bias and ψ is a kernel function.

Commonly used kernel functions are: linear $x_i^T x$, polynomial with degree d $(x_i^T x + 1)^d$, and radial basis function $\exp(-|x - x_i|_2^2/\sigma^2)$.

The lines that separate the data are defined by:

$$\omega^T \varphi(x_i) + b \geq 1, \text{ for } y_i = 1 \tag{2}$$

$$\omega^T \varphi(x_i) + b \leq -1, \text{ for } y_i = -1 \tag{3}$$

This is equivalent to the non-linear function:

$$y_i[\omega^T \varphi(x_i) + b] \geq 1, \text{ for } i = 1, \dots, n \tag{4}$$

This maps the input data to a high dimensional space and finds the hyperplane that perfectly separates the classes.

2.1.3. Naïve Bayes

Naïve Bayes is a simple probabilistic classification algorithm that classifies samples according to the likelihood of occurrence [21]. It assumes that features are independent given the class. During training, probabilities are calculated for each feature value given a class label. These probabilities are used to predict the label of a test sample.

Consider a feature vector, $\boldsymbol{X} = (X_1, \dots, X_n)$, where each feature value is taken from a distribution D_i. The set omega contains all feature vectors: $\Omega = D_1 \times \dots \times D_n$. Let C be the class label of an example.

The class posterior probabilities given a feature vector can be defined as a discriminant function: $f_i^*(\boldsymbol{x}) = P(C = i|\boldsymbol{X} = \boldsymbol{x})$. This can be rewritten after applying Bayes rule: $\mathrm{P}(\mathrm{C} = \mathrm{i}|\mathbf{X} = \mathbf{x}) = \frac{\mathrm{P}(\mathbf{X}=\mathbf{x}|\mathrm{C}=\mathrm{i})\mathrm{P}(\mathrm{C}=\mathrm{i})}{\mathrm{P}(\mathbf{X}=\mathbf{x})}$. Here, $\mathrm{P}(\mathbf{X} = \mathbf{x})$ is the same for all classes and can be eliminated. Thus, Bayes discriminant functions can be written as the following: $f_i^*(\boldsymbol{x}) = P(\boldsymbol{X} = \boldsymbol{x}|C = i)P(C = i)$, where $\mathrm{P}(\mathbf{X} = \mathbf{x}|\mathrm{C} = \mathrm{i})$ is termed as the class-conditional probability distribution.

Finally, the Bayes classifier can be defined as:

$$h^*(\boldsymbol{x}) = argmax\ P(\boldsymbol{X} = \boldsymbol{x}|C = i)P(C = i) \tag{5}$$

$h^*(\boldsymbol{x})$ finds the maximum a posteriori probability for any example x. Extending (5) to simplified naïve Bayes assumption that features are independent given class, we get the following form:

$$f_i^{NB}(\boldsymbol{x}) = \prod_{j=1}^{n} P\left(X_j = x_j \middle| C = i\right)P(C = i) \tag{6}$$

2.1.4. K-Nearest Neighbor

One of the classical and simplest nonparametric classification algorithms is the k-nearest neighbor (KNN) classifier, which classifies new examples based on nearest sample observation. It is based on the assumption that when feature vectors for training data points are projected into a subspace, any new data point can be classified based on its proximity to its k nearest neighbors [22].

Consider a set of n training samples, $x_1, \dots, x_n$, each having a label from a set or r labels, $Y = y_1, \dots, y_r$, and m features. The feature vector for x_i is represented as $x_{im}, \dots, x_{nm}$. A new sample s is assigned label y_i if a majority of k nearest neighbors of s possess the label y_i.

Nearness can be measured using any of the several distance measures. The most common ones are Euclidean distance (L2 norm), Manhattan distance (L1 norm) or Max norm.

The Euclidean distance between two samples x_a and x_b is defined as:

$$D(x_a, x_b) = \sqrt{(x_{a1} - x_{b1})^2 + \cdots + (x_{am} - x_{bm})^2} \tag{7}$$

The number of nearest neighbors in the neighborhood, k, is usually tuned as a hyperparameter. Empirically, as k increases, the accuracy of the prediction decreases. Fig. 7 presents a feature space with positive and negative labels and a test point where $k = 3$. Based on majority voting, the test point will be classified as negative.

Several variations of KNN exist in the literature. Weighted KNN adds weight to the vote of each label in the neighborhood based on its distance from the test sample [23]. Epsilon-ball KNN is a method that selects neighbors within a distance from the test sample.

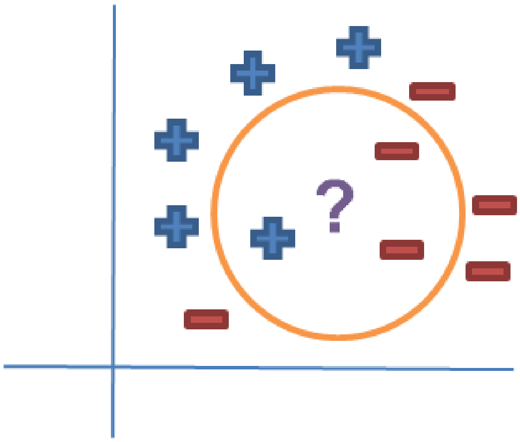


Figure 7. KNN with $k = 3$

2.1.5. U-Net

In 2016, Ronneberger et al. [24] proposed an extended fully connected convolutional network called U-Net intended to segment biomedical images. The network utilizes a novel training strategy based on the efficient use of data augmentation through elastic deformations. In other words, the learning process does not require a magnanimous amount of training samples to produce a good output. In addition to using the training data efficiently, it offers two more modifications. First, the network consists of an up-sampling stage and a down-sampling stage. During the up-sampling stage, a number of feature channels are created which help in transferring context information to the high-resolution layers. Instead of incorporating fully connected layers, the network uses an overlap tile strategy for predicting the pixels in the border region. The overlap tile strategy mirrors the pixels in the border region to provide more context information. Second, they introduce the application of weighted loss for separating connected structures.

The U-Net architecture, shown in Fig. 8, contains a total of 23 convolutional layers. It is composed of three parts: a contractive path (down-sampling), a bottleneck and an expansive path (up-sampling). The network consists of ten consecutive sets of the following: two 3×3 convolution layers each followed by a rectified linear unit (ReLU). The first four sets comprise the contractive path, and each set is followed by a max pooling operation. The number of feature channels are doubled after each max pooling operation. The part between the contractive and expansive is the bottleneck which consists of two convolution layers and a dropout. The next four sets comprise the expansive path. The process mentioned in the contractive path is mirrored in the expansive path. The convolution layers are preceded by a deconvolution layer with stride 2 and concatenation with the corresponding cropped feature maps from the layers in the contracting path. The last layer is a 1×1 convolution layer.

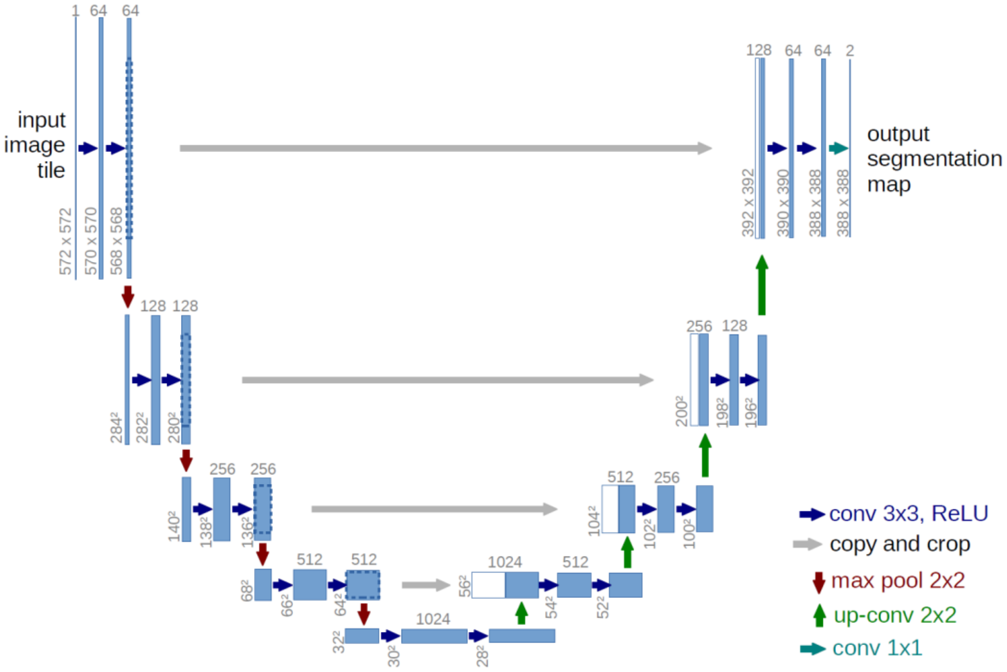


Figure 8. The U-Net architecture [24]

2.1.6. GoogleNet

GoogleNet is a deep convolutional neural network trained on images from a subset of the ImageNet database, which contains about 1.2 million images. The network is trained to distinguish between 1000 classes. The network is composed of 22 layers and is shown in Fig. 9. The input size to the network is 224×224×3 image. The network uses a number of Inception modules, which is a set of 1×1, 3×3 and 5×5 convolutions, and 3×3 max pooling. Each of the convolutional layers in the Inception module are followed by a ReLU non-linearity function. To combat overfitting, a number of 1×1 convolutions and average pooling are used. The 1×1 convolutional unit reduces the number of computations and reduces the dimensionality of the network in a non-linear way. The network forces dropout in the initial layers and towards the end before the fully connected layer. The fully connected layer is fed to a 1000-way softmax for classification.

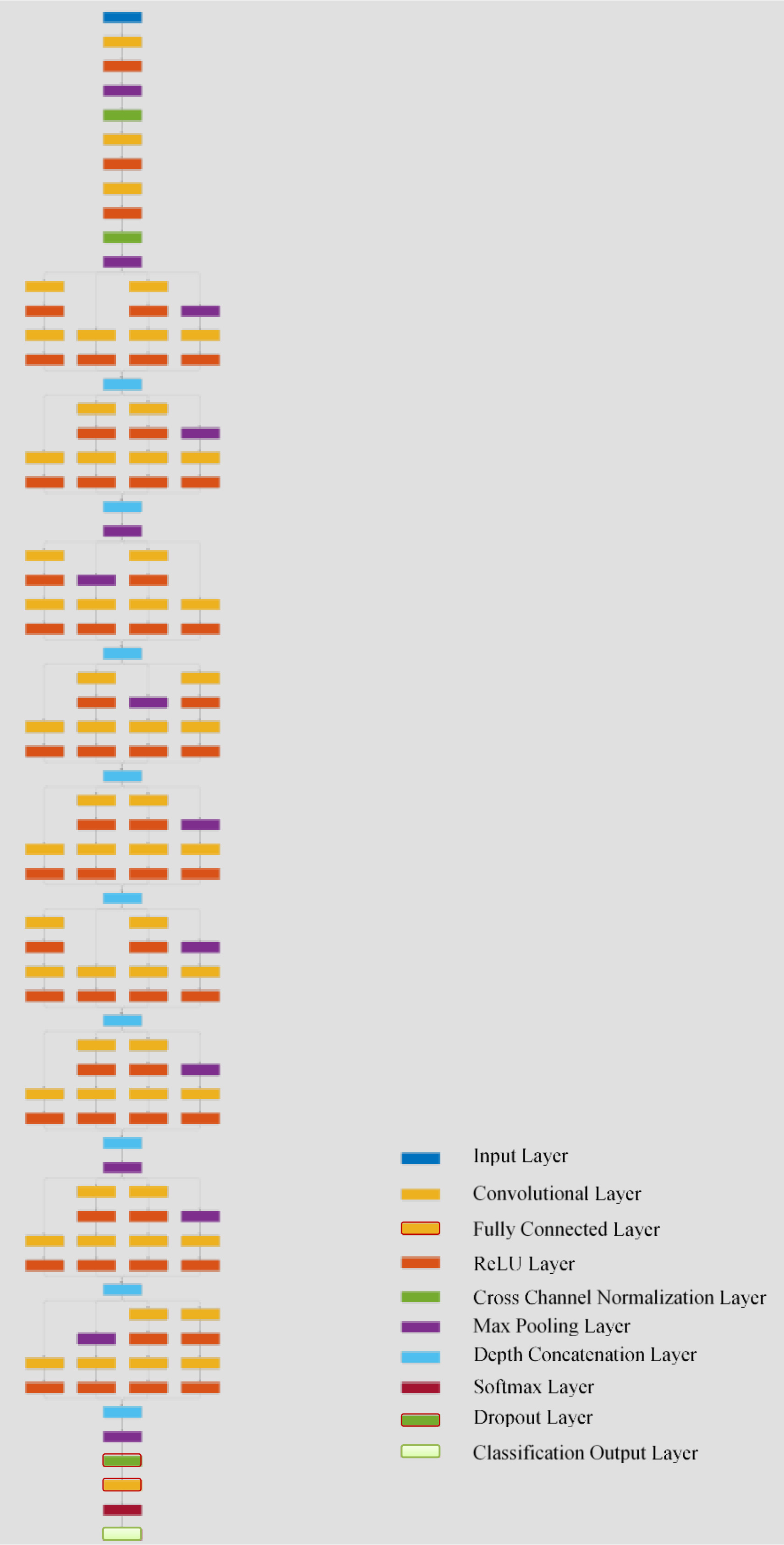


*Figure 9. GoogLeNet architecture**

**Network flow is top-down.*

2.2. Image Processing Techniques for Segmentation of Curvilinear Structures

As CCM is progressing towards establishing itself as a biomarker for detection of preliminary neuropathy, the need for systems facilitating completely automatic image analysis, neuropathy severity prediction and disease prediction is becoming inevitable. The establishment and recognition of new and reliable standards for nerve measurements can be achieved through correct segmentation and quantification of nerves in reasonable time. We found that the problem of nerve segmentation in corneal images holds a resemblance to vessel segmentation in fundus images, vascular structures in angiographic images, vein segmentation in leaf images, and fingerprint segmentation in fingerprint images. All of these problems mentioned fall under the category of identification of curvilinear structures. Consequently, techniques proposed for one of these problems can be exploited to resolve another. In the following subsections we review the existing literature on image processing techniques, mainly for corneal nerve segmentation and retinal vessel segmentation.

2.2.1. Region of Interest Expansion

A study of related literature has revealed many research groups that have been developing successful methods for automating the segmentation of corneal subbasal images. Ruggeri et al. [25] proposed a nerve recognition and tracing method for corneal images. Their algorithm was a modification of the one originally devised for vessel segmentation in retinal images [26]. The images first undergo a denoising process to enhance image contrast. This is followed by a nerve tracking process based on region growing. It involves extraction of seed points from the image using a uniformly spaced grid of 10 pixels. The expansion of the region of interest (ROI) is carried out by matching neighboring pixels according to a preset pixel intensity similarity threshold. On the encounter of a nerve intersection, a technique called bubble analysis is applied. The bubble analysis algorithm identifies nerve pixels occurring in the path of concentric circles drawn from the point of intersection. The next step is to classify pixels as belonging to the nerve or the background, using fuzzy c-mean clustering. There is a possibility of termination of the nerve tracking process resulting in discontinuous nerve segments. This issue is addressed by a nerve continuity algorithm which connects disjoint nerve segments. In order to assess the performance of the algorithm, 12 images from the subbasal nerve plexus of the cornea, obtained via a slit-lamp scanning CCM, were segmented using the proposed algorithm. The segmentation algorithm showed the tendency of generating false positives due to the presence of cells in the image background. However, this problem may have risen only because the images were obtained via slit-lamp CCM. Images using the state-of-the-art laser scanning CCM eliminate much of the image artifacts and is capable of producing images with better resolution. The time required for the segmentation of each image was approximately 4 - 5 minutes. Scarpa et al. [27] used a similar algorithm but introduced the use of a Gabor filter prior to the nerve tracking procedure. They evaluated the algorithm on a set of 90 images taken from the corneas of control subjects and patients. The technique succeeded in correct identification of more than 80% of the nerve length as compared to manual segmentation. Poletti and Ruggeri [28] further improved the nerve tracking algorithm presented in [25] by identifying seed points through multiple orientations of lines. Thereafter, nerves were segmented from the background by connected seed points identified from Dijkstra’s shortest path algorithm. The proposed algorithm was tested on a set of 30 CCM images. Single image processing time was reduced from 4 minutes to 25 seconds.

2.2.2. Gabor Wavelets

A significant contribution in this domain is by Dabbah et al. [16]. Their proposition of a dual model algorithm for nerve fiber detection was a revolutionary attempt to address the problem. Their dual model consists of Gabor wavelet filter and Gaussian envelope. They enhanced the nerve fibers in the image through the use of 2D Gabor wavelet filters for nerve identification and applied a Gaussian filter for background noise removal. The process of nerve fiber orientation estimation involved calculating distance through least mean squares (LMS) algorithm. This is followed by passing the image through a low pass Gaussian filter which results in diminishing the texture present in the background. They presented a comparative analysis of the Gabor method and a previously implemented Linop method (linear operator for detecting asbestos fibers in mammograms [29]), on a small dataset of 12 images. Results showed an improved performance by the Gabor method and a lower estimated error rate (EER) [30]. However, experiments conducted on such a small dataset fail to generalize the performance of the algorithm on large scale.

2.2.3. Morphological Operations

Recently, Al-Fahdawi et al. [31] designed a complete automatic system for nerve quantification based on morphological operations. The first part is nerve contrast enhancement which uses a combination of coherence and Gaussian filters for background noise reduction. Then, morphological operations of dilation, erosion, opening and closing are applied to remove stray and unwanted segments from the nerve structure. This is followed by edge detection using Canny edge detector for identifying the nerve fibers in the image. Their choice of edge detection is based on the fact that Canny edge detector is known for accurate filtering of multiple responses from a single edge. The work also presents a new algorithm for appropriate linking of gaps in the fibers. The technique is as follows. The image is converted to its skeleton and endpoints of each segment are identified and marked with a binary circle. Based on the distance between two disconnected endpoints, a straight line connects two disjoint endpoints. Finally, the image is thinned again, which results in shrinking the circles to a line. The segmented binary image is then used to extract clinical features namely, nerve tortuosity, length, density and thickness. The performance of the proposed system was evaluated on two datasets, having 498 and 919 images respectively, captured from a laser CCM. Single image processing time is approximately 7 seconds. They used non-conventional evaluation metrics of structural similarity index (SSIM), variation of information (VOI) and probability random index (PRI). These metrics have not been used elsewhere to evaluate nerve segmentation, and their method lacks a comparison with other methods in the literature.

2.2.4. Pixel Classification using Machine Learning

Pixel classification for CCM images has also been approached through supervised machine learning techniques. In this method, each pixel is considered a sample with the corresponding label of nerve/non-nerve (for nerve images) or vessel/non-vessel (for retinal images), and a feature vector extracted using pixel-based information. Thus, this becomes a simple binary classification problem. A multiscale enhancement of the dual model in [16] is presented in [15], which classifies pixels as nerve or non-nerve by training neural network or random forest classifiers. The evaluation of the trained model resulted in the best sensitivity and specificity at an EER of 15.44%. Guimaraes et al. [32] proposed a nerve segmentation system which involved morphological operations and machine learning for nerve classification. First, they used top-hat filtering for enhancing the image contrast, computed by subtracting the morphological opening of the image from the image itself. This is followed a nerve enhancement procedure using log-Gabor filters. Then, nerve segmentation is carried out through hysteresis thresholding. This results in a number of candidate nerve segments which may be true positives or false positives. To differentiate between them, they trained a Support Vector Machine with radial basis kernel function. They tested the efficiency of their approach by classifying pixels of 246 CCM images using Support Vector Machine (SVM), achieving an average sensitivity of 88%.

For retinal fundus scans, Rani et al. [33] proposed a retinal vessel segmentation approach using matched filter designs. They selected the green channel of the images for the segmentation process. Image preprocessing stage includes the application of contrast limited adaptive histogram equalization on the images. They applied a Gaussian based matched filter for the detection of vessel structures in the image. This results in the extraction of non-vessel structures as well, which is handled by the following steps. Using pixel-based features from each connected component, SVM and Tree-bagger is applied to classify components as vessels or not vessels. They reported an accuracy of 95% on the STARE database and 94% on the DRIVE database.

2.2.5. Hessian Matrix-based Approach

Jerman et al. [34] proposed a novel enhancement filter for segmenting vascular structures that occur in angiographic images. The multiscale enhancement function is derived from a Hessian matrix filter and can be adapted for both 3D and 2D images. The resultant function yields a uniform response to varying contrast values of the image. Hessian eigenvalues are defined for bright structures appearing on dark background. The Jerman enhancement function is a ratio of eigenvalues with response values ranging from 0 to 1. The enhancement filter was evaluated on 3D and 2D images of lung, cerebral and retinal vasculatures and demonstrated consistent results. The results were highly comparable results to other functions proposed in the literature.

2.2.6. Pixel Classification using Convolutional Neural Networks

There has been a plethora of studies in the literature that use CNNs for image segmentation. Fu et al. [35] used deep learning architecture for retinal vessel segmentation. They proposed a fully connected CNN for learning the vessel features followed by a fully connected conditional random field for converting the coarse probability map to binary. They related the curvilinear segments detection problem to a boundary detection problem and used a holistically nested edge detection (HED) technique proposed elsewhere [36] and based their implementation on this HED. Their deep learning architecture consists of 5 stages where each stage consists of several convolutional and ReLU layers. The last convolutional layer in each stage is linked to the side output layers. Two datasets were used for evaluation of the proposed technique. They achieved sensitivity of 73% and 71% on the DRIVE and STARE datasets respectively. They showed that although their method did not produce extraordinary results when compared with other techniques, they did succeed in reducing the false positive in the optic disc and pathological regions.

The concept of using U-Net [24] for corneal nerve segmentation was introduced by Colonna et al. [37]. U-net is a convolutional neural network intentionally designed for segmentation of biomedical images [24]. It has successfully been applied for segmentation of brain tumors in MRI scans [38], generating synthetic images by mapping retinal vessel trees to retinal fundus scans [39], cell membrane segmentation in electron microscopy images [36], and many others. Colonna et al. [37] trained a U-Net based CNN on CCM nerve images obtained from healthy and diabetic patients. In the preprocessing stage, the images were cropped of 10 pixels from the edges and resized to the input size required by U-Net using a bicubic transformation. The training was performed on 8909 images, out of which 30% was used for validation only. The network was trained for 6 epochs. Instead of using manual tracings for ground truth, they obtained segmented images by applying a previously proposed algorithm for segmentation. They evaluated the trained model on 30 test images and compared against manual tracings. In order to allow for slight shifts in the nerve position, a tolerance of 3 pixels was allowed. Results revealed a sensitivity of 97% and a false detection rate of 18%. However, since the number of test images is so small, the ability of the model to generalize cannot be determined.

Moreover, Son et al. [40] employed U-Net as part of a generative adversarial network (GAN) to segment vessels from retinal fundoscopy images. In order to get sharp segmented structures, they experimented with various discriminators. They were able to reduce false positives but with some trade-off in the number of false negatives. They reported a precision and recall of 91% on DRIVE and STARE datasets.

Maninis et al. [41] also provide a solution to the problem of retinal image segmentation using CNN. They improvise a CNN architecture by combining the architectures of VGG-18 and Inception V3. The fully connected layers towards the end of the VGG-18 architecture were removed and specialized convolutional layers to the end of each stage were added. The final result is formed by adding another convolutional layer to the end of the model.

2.3. Classification Techniques for Retinal and Corneal Images

Recently, convolutional neural networks have gained distinguished popularity among data scientists and it has been proven to solve a multitude of image-based problems including segmentation, localization and classification. Specially with the introduction of the concept of transfer learning using pretrained networks, such as GoogleNet [42], AlexNet [43] etc, the requirement of a large amount of data to train the neural network ceases to be necessary. For this reason, pretrained CNNs have become the quintessence of deep learning. To the extent of our knowledge, not much work has been done on neuropathy classification of corneal images. Therefore, since the corneal nerve images hold an apparent visual and conceptual similarity to the retinal images obtained using fundoscopy, the following subsections explore the literature for retinopathy classification techniques applied on retinal fundus scans, in addition to the scarce literature on classification of corneal nerve images. Several approaches have been proposed in the literature for classification of retinal images using transfer learning and fine-tuning pretrained CNNs. Others have tackled the problem using custom-designed deep learning models.

2.3.1. Transfer Learning and Parameter Tuning

Research work in the literature shows many evidences of successful usage of pretrained models using transfer learning and fine tuning. The use of pretrained CNNs have become widespread due to their ease of use. Moreover, they do not require millions of training samples to produce a well-trained model, which is the case for networks trained with random initialization.

Gulshan et al. [44] used transfer learning for a detailed and multi-aspect classification of retinal images obtained from fundus photography. They classified images as (a) having referable diabetic retinopathy (mydriatic, non-mydriatic or both), (b) into subtypes of diabetic retinopathy (moderate or worse only, severe or worse only, diabetic macular edema only) and (c) as gradable or non-gradable. Training was conducted on the pretrained model of Inception-v3 architecture. They fine-tuned an Inception V3 model and retrained it on retinal scans using the initial weights provided by the pretrained model. For an assessment of their approach, they conducted tests on 2 datasets and reported a sensitivity of approximately 90% for each classification type. Similarly, Choi et al. [45] employed transfer learning and fine-tuned VGG-19 and Alexnet architectures for classifying normal retinal images and 9 types of retinal pathologies. Their 10-class classification resulted in an accuracy of 58%. They compared this approach with commonly used machine learning classifiers (SVM, Random Forest, ANN, etc.) and found that these approaches were outperformed by CNN transfer learning method. Moreover, even with deep learning, when moving from binary to multiclass classification, the accuracy decreases as the number of classes increases.

Lam et al. [46] attempted to perform multiclass classification of the severity levels of retinopathy from retina fundus scans. They evaluated the usage of different optimizers, learning rates, gradient update policies and dropout levels for fine tuning pretrained Googlenet and Alexnet architectures. Experiments were conducted on the publicly available Kaggle dataset of 35000 retinal images categorized into varying levels of retinopathy (class labels: normal, mild, moderate, severe, end stage), and on the Messidor dataset (class labels: normal, mild, moderate, severe). They reported the best accuracy of 74% on Messidor dataset obtained when augmented dataset was fed to the model for training. On the Kaggle dataset, they scored an accuracy of 84%.

Burlina et al. [47] employed pretrained Overfeat for extracting features from retinal fundus scans classification. Overfeat [48] is a pretrained model, similar to GoogleNet, trained on images from ImageNet. It implements a multiscale and sliding window approach inside a convolutional network and applies the concept of combining multiple localized predictions instead of focusing on the background. The problem is classified as four stages of retinopathy. They subdivided the problem into multiple binary classifications: stage 1,2 vs 3,4; stage 1,2 vs 3; stage 1 vs 3; stage 1 vs 3,4. Features extracted from Overfeat were fed into a linear SVM for classification. For all stages, an accuracy of greater than 90% was achieved. Continuing their experiments, Burlina et al. [49] compared the previous approach with AlexNet trained from scratch.

Experimental results on the AREDS fundus image data showed that the Alexnet model (accuracy 91%) outperformed Overfeat (accuracy 84%). The reason for discrepancy in reporting the accuracy measures for Overfeat in [47] and [49] is not mentioned. However, this might be because the latter experiments conducted evaluations using cross validation.

Poplin et al. [50] trained an Inception-v3 for prediction of cardiovascular risk factors from retinal fundus images. They trained a binary classification model for predicting the risk factors of smoking status or gender, another classification model for predicting major adverse cardiovascular event (MACE) and a regression model for predicting age, body mass index (BMI), systolic blood pressure (SBP), diastolic blood pressure (DBP) and HbA1c. the results were compared against the accuracy of a random classifier. For HbA1c, BMI and DBP, the model prediction was almost similar to that of a random classifier. However, for age and SBP, the trained model's predictions were reported as 78% and 72% respectively and showed much improvement against random classification. They reported an area under the curve (AUC) of 97% and 70% for gender and MACE classifications. These results were reported using the UK Biobank dataset.

2.3.2. Combination of Pretrained Models and Ensembles

Ting et al. [51] adapted the VGG network and trained eight different CNNs for different kinds of outputs contributing to the final score. Three ensembles of two CNNs each was used for classification of retinal images into (a) retinopathy severity levels, (b) AMD scores, and (c) glaucoma levels. The two networks are trained using original images and contrast-normalized images respectively. The final score for each category is the average of the scores from their respective pair of classifiers. Two more CNNs are used for determining whether the score is ungradable of nonretinal. The final scores from each category formulate the decision for referable diagnosis.

Abramoff et al. [52] presented an automated device for detection of diabetic retinopathy (DR). They developed a CNN architecture inspired by AlexNet and VGG networks. Their classifier produced outputs indicating: (a) presence of diabetic retinopathy, (b) presence of referable DR, (c) vision threatening DR, and (d) quality of image or examination. Image augmentation included rotational, spatial and scale augmentations. The output from the CNN produces a probability implying that whether a detection is an abnormality. These probabilities form a feature vector which is supplied to two fusion algorithms. The two algorithms predict whether the image represents referable DR or vision threatening DR respectively. The system was evaluated on the Messidor-2 dataset for retinal fundus images. A sensitivity of 96% was reported for the first output, and 100% for the rest.

2.3.3. Modification of Available Neural Networks

Colonna et al. [37] proposed the idea of neuropathy classification of CCM nerve images using a CNN inspired from U-Net in addition to nerve segmentation. The existing U-net architecture was modified for image-based classification as normal or neuropathy. The features extracted from the lower layers of U-Net were fed to a convolution layer, followed by max pooling and fully connected layers for final classification. They trained this modified network on 5000 images, using 30% of it as validation. The network was trained with a batch size of 256 images for 15 epochs with shuffling enabled before every epoch. The optimization method was set to stochastic gradient descent with L2 regularization and a learning rate of 0.01. The performance of the trained model was evaluated by predicting labels on 100 unseen images. They succeeded in classifying with an accuracy of 83%. However, further insight on the evaluation results is not provided.

Takahashi et al. [53] performed a thorough study on two types of classifications for retinal images. The first is a 3-class classification for the grades of retinopathy (simple diabetic retinopathy; pre-proliferative diabetic retinopathy; proliferative diabetic retinopathy) using a modified version of the GoogleNet architecture. The top five layers of GoogleNet were removed and some parameters were fine-tuned. The network was trained on more than 9000 images. An accuracy of 81% was scored for the prediction of retinopathy grade. They trained another model using the same network design and trained it on images from patients with follow-ups to predict their prognosis. The prognoses consisted of multistage outputs. The first output was to predict whether a treatment was required and if required then whether it has to be in the current visit or the next. The second output predicted one of the three possible treatments. The third output was to determine the visual acuity of the image as stable, improved or worsened. They achieved an accuracy of 96% with a low false negative rate which implied few mispredictions when the treatment was not required but the network predicted otherwise.

2.3.4. Custom Designed Convolutional Neural Networks

Tan et al. [54] introduced the idea of detection of age-related macular degeneration on retinal images using deep learning. Their CNN architecture consisted of 14 layers including 7 convolution layers, 4 max pooling and 3 fully connected layers. They trained the model using blind and 10-fold cross validation on private dataset. Prior to training, data augmentation was performed by rotation. For optimization, the Adam optimizer was used. They achieved a sensitivity of 93% and a specificity of 88% for blind cross validation and an average of 96% sensitivity and 94% specificity for 10-fold cross validation on the AREDS dataset.

Gargeya and Leng [55] designed a convolutional neural network based on the principles of deep residual learning for the classification of retinal color fundus images into normal or retinopathy. During data preprocessing, image brightness levels were adjusted, and data augmentation was applied to allow rotation invariant predictions. The CNN network was designed such that each convolutional layer combined the output from itself and previous layers. The authors give an abstract understanding of the network and do not provide the details. A visual heatmap was also developed to visualize the severe retinopathy regions in the images. The CNN was used to construct a feature vector of 1024 features from each image, which was combined with the meta-data features and then fed to a decision tree classifier for final prediction. Experiments were conducted using 5-fold cross validation. A sensitivity of 93% and a specificity of 87% was achieved on the Messidor-2 dataset. Compared with other approaches, a very high specificity was achieved using this method.

2.3.5. Machine Learning

Silva et al. [56] approached neuropathy classification of corneal images using SVM classifiers. They extracted a vector of 61 texture-based features from the images which was reduced to 6 features after applying principal component analysis (PCA). SVM was used to classify images as (a) with or without neuropathy, and (b) mild or moderate neuropathy. Using 10-fold cross-validation an accuracy of 73.5% was achieved for the first classification and 79.3% for the second classification on the dataset used in [57]. Although the total number of images is large (631 images), these images were acquired from 20 subjects only, so the dataset lacks image diversity from multiple patients. Moreover, the acquired images are overlapping, which results in the same corneal region appearing in multiple images, contributing to the high value of accuracy.

Existing research shows that segmentation of curvilinear structures has been approached in many ways. This has been experimented extensively on corneal nerve images, retinal fundus scans, leaf images and other similar images. Table 1 and Table 2 summarize the segmentation techniques for curvilinear structures and classification techniques for retinal and nerve images respectively, that were presented in this research. To the best of our knowledge, neuropathy classification of corneal nerve images has not been approached by the research community except for two pilot studies [37], [56]. In [37], the authors do not give a detailed insight and discussion into the neuropathy classification results reported in their article. Moreover, the type of classification conducted is binary (healthy/pathological). In [56], the dataset contains largely overlapping images from a small number of patients, resulting in a lot of repetitive images. Therefore, the evaluation does not give us a clear indication of whether the classification model will generalize well. Thus, research in this area provides a lot of room for investigation on the idea of classification, extending the classification to multiclass, and exploring the potential of other machine learning algorithms to solve this problem.

Table 1. Summary of Segmentation Techniques for Curvilinear Structures

Method Classification	**Related Literature**
Region of Interest Expansion	• Poletti and Ruggeri [28]: Seed point extraction, fuzzy c-means clustering and nerve continuity algorithm, tested on 12 images, resulted in a large number of false positives; • Scarpa et al. [27]: Based on [24], introduced the use of a Gabor filters. Tested on 90 images. Correct

Method Classification	**Related Literature**
	identification of more than 80% of the nerve length; • Ruggeri et al. [25]: Improvement on [25] by identifying seed points through multiple orientations of lines, segmentation of nerves by connected seed points using Dijkstra’s algorithm for finding shortest path
Gabor Wavelets	• Dabbah et al. [15]: Proposition of a dual model consisting of Gabor wavelet filter and Gaussian envelope
Morphological Operations	• Al-Fahdawi et al. [30]: Nerve contrast enhancement using coherence and Gaussian filters, morphological operations of dilation, erosion, opening and closing are applied to remove stray and unwanted segments, edge detection using Canny edge detector, new method for linking gaps in the nerves
Pixel Classification using Machine Learning	• Rani et al. [32]: Retinal vessel segmentation using Gaussian based matched filter designs, selection of the green channel of the images application of contrast limited adaptive histogram equalization, using pixel-based features from each connected component; • Guimaraes et al. [31]: Top-hat filtering for enhancing the image contrast, nerve enhancement procedure using log-Gabor filters, nerve segmentation is carried out through

Method Classification	**Related Literature**
	hysteresis thresholding, pixel classification using SVM with RBF kernel; • Dabbah et al. [14]: A multiscale enhancement of the dual model in [16], classification of pixels as nerve/non-nerve by training NN or RF classifiers
Hessian matrix-based approach	• Jerman et al. [33]: Segmentation of vascular structures in angiographic images, a multiscale enhancement function is derived from a Hessian matrix filter
Pixel Classification using CNN	• Fu et al. [34]: A fully connected CNN for retinal vessel segmentation using HED technique proposed in [36]; • Colonna et al. [36]: Corneal nerve segmentation using U-Net based CNN, Tested on 30 images, achieved a sensitivity of 97% and FDR of 18%; • Son et al. [39]: Employed U-Net as part of a GAN for retinal vessel segmentation; • Maninis et al. [40]: Retinal image segmentation by combining the CNN architectures of VGG-18 and Inception V3.

Table 2. Summary of Classification Techniques for Curvilinear *Structures*

Method Classification	Related Literature
Transfer Learning and Parameter Tuning	• Gulshan et al. [44]: Inception V3 architecture, 3 types of classification: images as (a) having referable diabetic retinopathy (mydriatic, non-mydriatic or both) or not, (b) subtypes of diabetic retinopathy (moderate or worse only, severe or worse only, diabetic macular edema only) and (c) as gradable or non-gradable; • Choi et al. [45]: VGG-19 and Alexnet architectures for classification of normal retinal images and 9 types of retinal pathologies (10-class classification); • Lam et al. [46]: Multiclass classification of the severity levels of retinopathy from retina fundus scans using GoogleNet and AlexNet; • Burlina et al. [49]: Pretrained model, Overfeat, for retinal image classification and compared the results with AlexNet trained from scratch.
Combination of Pretrained Models and Ensembles	• Ting et al. [51]: Adapted the VGG network and trained 8 CNNs for different kinds of outputs contributing to the final score. Three ensembles of two CNNs each was used for classification of retinal images into (a) retinopathy severity levels, (b) AMD scores, and (c) glaucoma levels; • Abramoff et al. [52]: An automated device for detection

Method Classification	Related Literature
	of diabetic retinopathy (DR) inspired by AlexNet and VGG networks, the classifier produced outputs indicating: (a) presence of diabetic retinopathy, (b) presence of referable DR, (c) vision threatening DR, and (d) quality of image or examination.
Modification of available neural networks	• Colonna et al. [37]: Binary classification of CCM nerve images (healthy/pathological) using U-Net; • Takahashi et al. [53]: Modified version of the GoogleNet architecture, (a) 3-class classification for grades of retinopathy and (b) multistage classification of patients with follow-ups to predict their prognosis.
Custom-designed Convolutional Neural Networks	• Tan et al. [54]: CNN architecture for retinal image classification, consisted of 14 layers including 7 convolution layers, 4 max pooling and 3 fully connected layers; • Gargeya and Leng [55]: A CNN based on the principles of deep residual learning for binary classification of retinal images, the CNN produced a feature vector of 1024 features from each image, which was combined with the meta-data features and then fed to a decision tree classifier for final prediction.

CHAPTER 3: METHODOLOGY

We approach the classification of corneal nerve images using two different methods. The first method involves three steps: nerve segmentation, feature extraction and classification. The second method uses convolutional neural networks to classify raw images. Both of these methods are described in the following subsections.

3.1. Image Acquisition

We evaluated our techniques on two patient datasets provided by the medical research team at Hamad Medical Center (HMC), Qatar. The first dataset (named as DPN1) consists of confocal images of the subbasal nerve plexus from subjects diagnosed with type 1 diabetes. The second dataset (named as MSN1) consists of confocal images of the subbasal nerve plexus from subjects diagnosed with Multiple Sclerosis (MS). Images were captured by separate ophthalmologists for diabetes and MS. In addition to this, corneal images of healthy control subjects were also provided.

All images provided by HMC were acquired using laser scanning Heidelberg Retinal Tomograph (HRT-III) equipped with the Rostock Cornea Module (Heidelberg Engineering GmbH, Heidelberg, Germany). The helium neon diode laser source of this device uses a 670-nm red wavelength which does not pose any significant for ocular safety. Its 63× objective lens has a numerical aperture of 0:9 (Olympus, Tokyo, Japan) and covers a field of 400×400 μm^2. A focal depth of 40-60 μm was used to capture images from the subbasal nerve plexus layer of the cornea. Each of the obtained images has a size of 384×384 pixels and is saved in bitmap format. An anesthetizing gel was applied to the subject's cornea before bringing it in contact with the microscope lens. Fig. 10 shows a magnified representation of the subbasal nerve plexus of the human cornea and indicates the central region as well as the inferior whorl region. A number of non-overlapping images were captured from the central region of the cornea of each subject. 3 – 4 high clarity images were selected per eye.

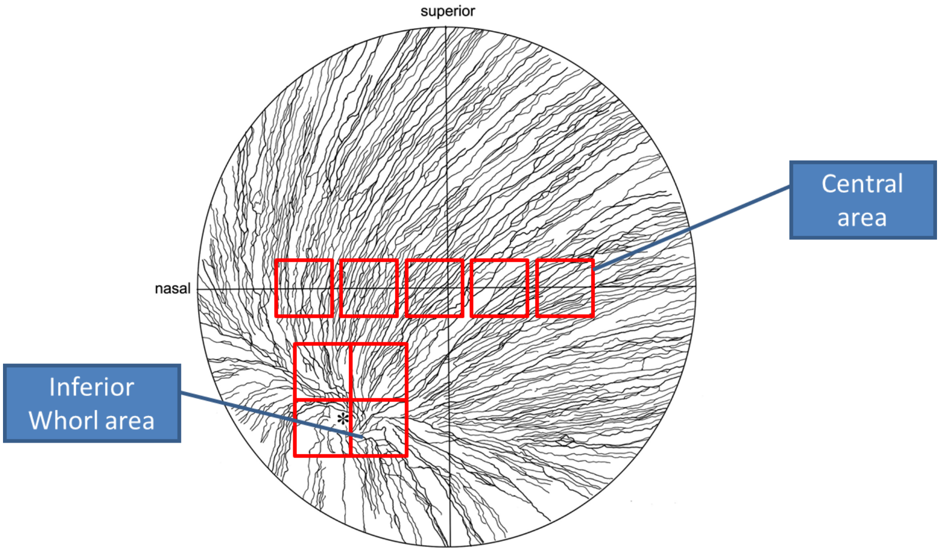


Figure 10. A magnified representation of the subbasal nerve plexus of the human cornea.

Additionally, another dataset was used for comparative analysis [57] (named as DPN2). Images were captured in a similar manner using the same model of the microscope. The subjects are healthy controls and diabetic patients. The image capture process was carried out in the University of Coimbra, Portugal. However, the images are not limited to the central region of the cornea and contain overlapping regions of the subbasal nerve plexus. Table 3 summarizes the subjects' details for each dataset.

Table 3. Subject Details for Each Dataset

	DPN1		**MSN1**		**DPN2**	
	controls	patients	controls	patients	controls	patients
Number(m/f)	21(10/9)	93(49/44)	0	51(16/35)	8	12
Age (years)	38.6(+/- 10.16)	47.5(+/- 16.27)	N/A	37.05 (+/- 9.59)	54 (+/-7)	58 (+/-10)

We have used the following naming notation to refer to the datasets in the experiments:

- DPN1: Images from controls and patients with diabetic peripheral neuropathy, provided by HMC (for segmentation and classification)
- MSN1: Images from patients having neuropathy due to Multiple Sclerosis, provided by HMC (for segmentation)
- DPN2: Images from controls and patients with diabetic peripheral neuropathy [57] (for segmentation and classification)

The number of images for the datasets considered for the classification problem are tabulated in Table 4. Images in MSN1 were not used for classification due to the fact that neuropathy culminates in different forms given different diseases and cannot be generalized. Since the appearance of diabetic neuropathy in the cornea has been well established in the corneal nerves by multiple studies we have trained and tested the classification model on diabetic images only.

Table 4. Actual Classification of Images in Each Dataset

	Healthy	**At-risk Neuropathy**	**Definite Neuropathy**	**Total**
DPN1	449	107	86	642
DPN2	48	28	25	101

3.2. Ground Truth Annotation

The images in DPN1 and MSN1 were manually traced by human experts at HMC. These tracings are used as ground truth for nerve segmentation. Using a proprietary, custom-built software, CCMetrics [15], the images in DPN1 are quantified for nerve loss. Based on the nerve fiber loss computed by the software, subject history as well as the doctors' experience in the field, the images were classified manually into three groups:

1. '*Healthy'*, indicating the status of images from the control subjects and some patients with less progression in the disease.
2. '*At-risk neuropathy*', indicating an early stage of neuropathy.
3. '*Definite neuropathy*', indicating a later stage of neuropathy.

The images in DPN2 were also manually traced and the tracings were used for comparing automated nerve segmentation. As stated earlier, DPN2 does not follow the same image capture protocol as DPN1 and MSN1 with regards to defining the image capture region of the cornea. Secondly, the neuropathy criteria defined for the central region of the cornea does not apply to the overall cornea. Therefore, for a fair comparison, only few representative images that belong to the central region of the cornea were selected from DPN2 for neuropathy classification. Furthermore, the neuropathy stages defined for diabetes do not apply to MS. Thus, MSN1 was eliminated from the neuropathy classification experiments.

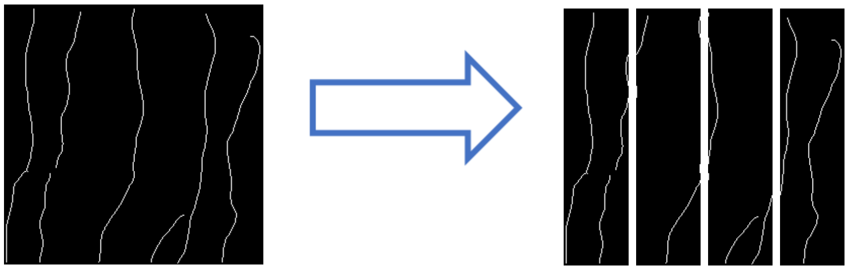
3.3. Nerve Segmentation Using Convolutional Neural Networks

As mentioned in section 2, segmentation of curvilinear structures has been approached through many techniques. However, many studies excluded images with artefacts [57], the most challenging one being the lens glare. In this study, we include images with lens glare in the background and show that convolutional neural networks have the capability to segment the nerves while effectively ignoring the lens glare. U-Net has been widely used in the literature for segmentation of medical images.

We trained a U-Net based CNN on a selection of our images and then used the trained model to predict the nerve pixels in the rest of the images. For the training process, images were carefully selected so as to train the model on all kinds of variations of corneal nerve images. The training images include images having any of the following artefacts: light reflection, structures other than nerves, pressure lines, background texture and faded nerves. The segmented images then undergo some post processing which includes binarization and skeletonization.

3.4. Feature Extraction

The final segmented images are used to extract features for machine learning classification. Research shows that the most important feature which corelates highly with diabetic neuropathy is Corneal Nerve Fiber Length (CNFL) [58]. CNFL is defined as the total nerve length in the image. Thus, we extract CNFL (referred as NFL in Section 4.2) from each image as a summation of all nerve pixels in the segmented image. The appearance of nerves differs in different sections of the image as neuropathy progresses and nerve damage is not necessarily present in all parts of the image. Therefore, intra-segment CNFL (referred as ISNFL in Section 4.2) would be a good set of features to extract. We split the image into 4 equal segments and calculate CNFL for each. Finally, we have a set of 5 features representing each image. Fig. 11 shows an example of how the images are sliced into segments.



(a) (b)

Figure 11. Image slicing example. (a) segmented image, (b) image slices to extract intra-segment CNFL.

3.5. Neuropathy Classification using Machine Learning

The feature set extracted from the previous step is used to classify the images into three classes; normal, at-risk neuropathy, or definite neuropathy. Adaptive Neuro-fuzzy Inference System (ANFIS) was chosen as the algorithm for classification due to the following two reasons. It combines neural networks and fuzzy logic and it can produce continuous output. The benefit of getting a continuous output is that it can be either converted to crisp by setting a threshold, or it can used as is to predict a severity measure. Furthermore, the classification is also performed using SVM, NB and KNN classifiers.

3.6. Hierarchical ANFIS Classification Model

There are two most common multiclass models for classification: one-versus-one (OVO; also called all-versus-all) and one-versus-all (OVA). We use one OVA and one AVA classifier to construct a multiclass model for ANFIS.

Our problem is multiclass with 3 classes: normal (0), at-risk neuropathy (1), definite neuropathy (2). The ANFIS classification model consists of implementation of two binary classifiers. The first classifier (C1) is trained to differentiate normal (0) from abnormal (1,2) samples (OVA). The second classifier (C2) is trained to classify between at-risk (1) and definite (2) neuropathy (OVO). For both classifiers, cut-off points are determined using the validation sets. Fig. 12 shows the hierarchical ANFIS classification process for training and validation.

Data samples are split into four partitions: one for training (T1), two for validation (V1, V2) and one for testing (T2). An initial fuzzy inference system (INFIS) is created using T1 and then supplied to ANFIS for training. Upon completion of the training, an output FIS is generated which can be (a) used for prediction or/and (b) used as an INFIS to retrain ANFIS. Using OUTFIS, output values are predicted on V1 and the cutoff point that gives the best accuracy is determined. To find the best cutoff point, a grid search method is used. Accuracy measures are calculated using an increment of 0.1 from 0 to 1 where (0) and (1) are the class labels. The cutoff point that gives the best accuracy is considered as the best cutoff point for this trained model. The next step is to retrain the ANFIS either using the same INFIS as before or the new FIS generated. This decision is made based on whether the accuracy has improved or not. The whole process is repeated n times. Once the maximum iterations are complete, either the best FIS can be used as the final prediction model or the FIS can be retrained on T1+V1. The set V2 is used to evaluate both of these models. One that gives a better accuracy among the two, using the cutoff point for the best model, is taken as the final model. This process is done for both classifiers, C1 and C2.

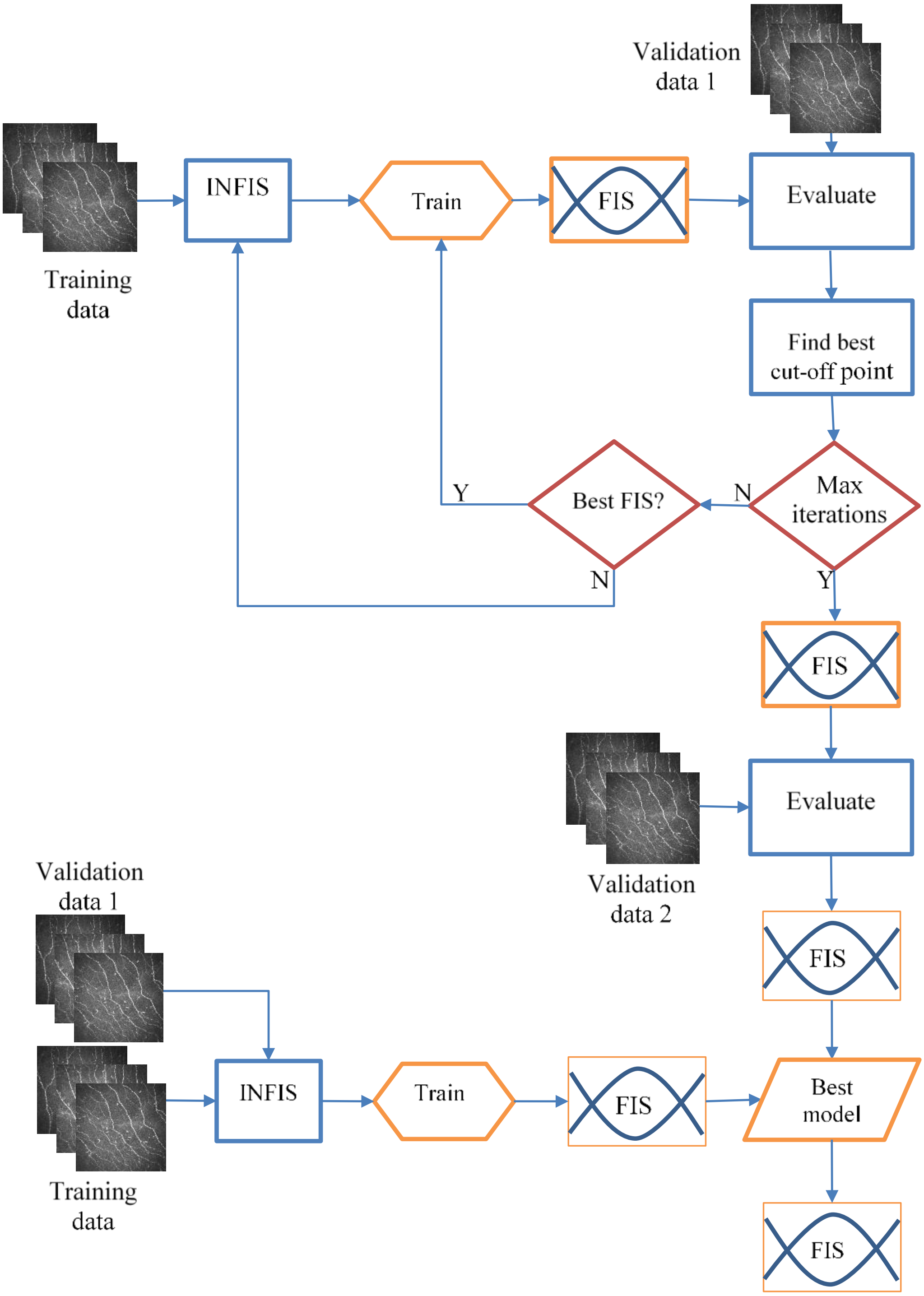


Figure 12 Hierarchical ANFIS classification model

During the testing phase, the class of an unseen sample is predicted first using C1. If the prediction is normal, this is considered as the final prediction, else C2 predicts its final class. Fig. 13 shows the prediction process using hierarchical ANFIS classifier.

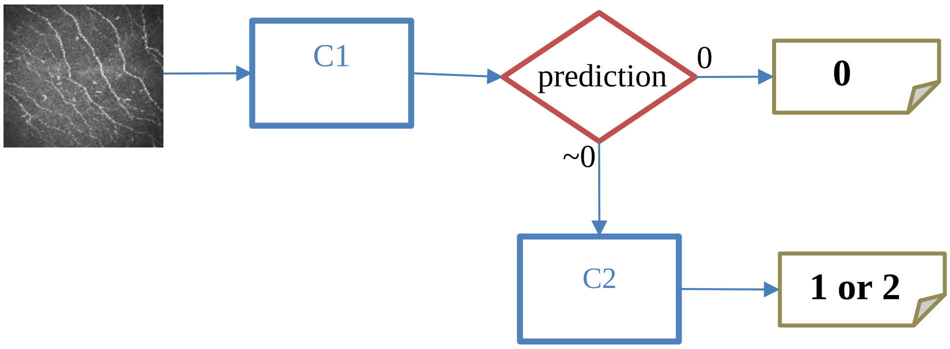


Figure 13. Hierarchical ANFIS prediction workflow

3.7. Neuropathy Classification using Transfer Learning

The classification of images is also conducted using deep neural networks. In this method, the feature extraction step is not required as raw images are fed to the network for classification. The literature review (Section 2.3) shows an extensive use of transfer learning approaches for classification of retinal fundus scans. In this research, GoogleNet is fine-tuned on the corneal nerves images to give the best performance. GoogleNet is a pretrained network on ImageNet which contains millions of images of everyday items classified into a thousand classes. The weights from the training on ImageNet are used as initial weights when the network is retrained on the corneal images. Additionally, optimizers, learning rate, momentum, batch size and epochs are fine-tuned to create the best classification model.

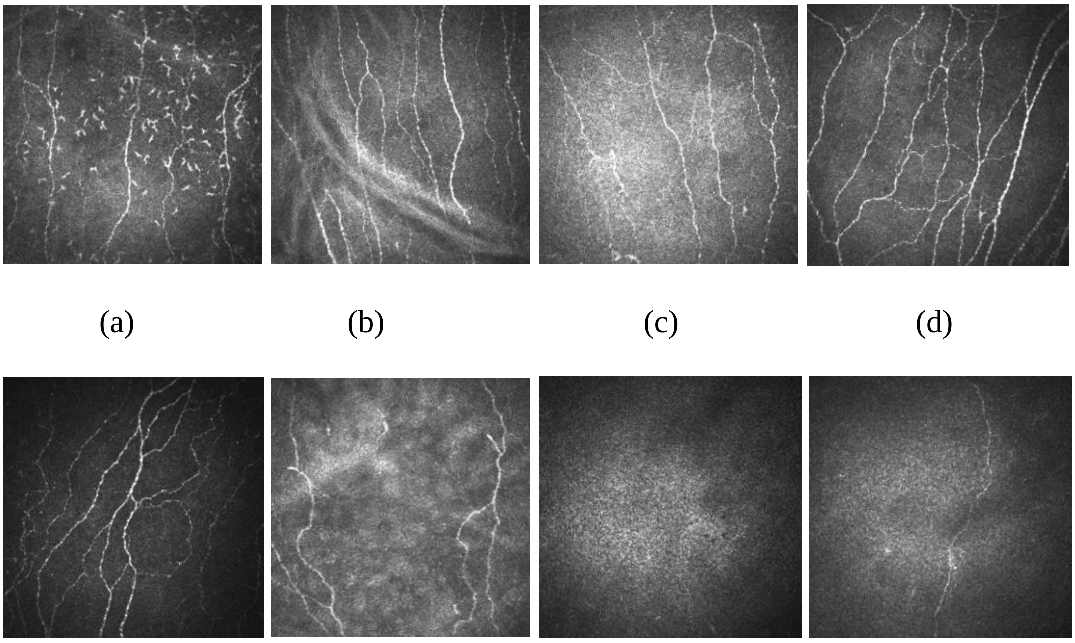
3.8. Experimental Procedures for Segmentation

In this section we describe the experimental procedure for CCM image segmentation including preprocessing, selection of training images, model training details and post-processing procedure.

3.8.1. Preprocessing and Selection of Training Images

The training images and their corresponding ground truth images (384×384) are resized to 256×256 pixels. A bicubic resizing operation resulted in the distortion of nerves in the binary ground truth images therefore a center crop was preferred for resizing to the required dimensions. However, for testing, the trained model accepts other sizes, therefore the images are segmented without any resizing. The output is a 256×256 image.

Instead of random selection, a careful selection of training images is carried out which includes images with different characteristics to increase the learned model's ability of classifying all kinds of corneal images correctly. Images are hand-picked from subsets DPN1 and MSN1 that represent the artefacts from the available data. Some of the selected images for training are displayed in Fig. 14.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 14. Images with artefacts selected for training. (a) images with structures from neighboring layers, (b) images with a pressure line, (c) images with nerves and overall glare, (d) images with nerves and no glare, (e) image containing partially faded nerves without glare, (f) image with background texture and glare streaks, (g) image with no nerves and center glare, (h) image with very few nerves and center glare.

3.8.2. Training

The U-Net based CNN is trained on the training images for 50 epochs using random initialization of weights. Learning rate is set to 0.0001 with Adam optimization and the loss function of binary cross entropy is used. The training time on an Intel Core i3-6100 Processor was approximately 30 hours. Implementation was done in python using the Scikit-Learn [59] and Tensorflow package [60] with the Keras wrapper. Table 5 provides a model summary of the network.

Table 5. Model Summary of the U-Net Based CNN

Layer (type)	**Output Shape**	**Param #**	**Connected to**
input_1 (InputLayer)	(None, 256, 256, 1)	0	
conv2d_1 (Conv2D)	(None, 256, 256, 64)	640	input_1[0][0]
conv2d_2 (Conv2D)	(None, 256, 256, 64)	36928	conv2d_1[0][0]
max_pooling2d_1 (MaxPooling2D)	(None, 128, 128, 64)	0	conv2d_2[0][0]
conv2d_3 (Conv2D)	(None, 128, 128, 128	73856	max_pooling2d_1[0][0]
conv2d_4 (Conv2D)	(None, 128, 128, 128	147584	conv2d_3[0][0]
max_pooling2d_2 (MaxPooling2D)	(None, 64, 64, 128)	0	conv2d_4[0][0]
conv2d_5 (Conv2D)	(None, 64, 64, 256)	295168	max_pooling2d_2[0][0]
conv2d_6 (Conv2D)	(None, 64, 64, 256)	590080	conv2d_5[0][0]
max_pooling2d_3 (MaxPooling2D)	(None, 32, 32, 256)	0	conv2d_6[0][0]
conv2d_7 (Conv2D)	(None, 32, 32, 512)	1180160	max_pooling2d_3[0][0]
conv2d_8 (Conv2D)	(None, 32, 32, 512)	2359808	conv2d_7[0][0]
dropout_1 (Dropout)	(None, 32, 32, 512)	0	conv2d_8[0][0]
max_pooling2d_4 (MaxPooling2D)	(None, 16, 16, 512)	0	dropout_1[0][0]
conv2d_9 (Conv2D)	(None, 16, 16, 1024)	4719616	max_pooling2d_4[0][0]
conv2d_10 (Conv2D)	(None, 16, 16, 1024)	9438208	conv2d_9[0][0]
dropout_2 (Dropout)	(None, 16, 16, 1024)	0	conv2d_10[0][0]
up_sampling2d_1 (UpSampling2D)	(None, 32, 32, 1024)	0	dropout_2[0][0]
conv2d_11 (Conv2D)	(None, 32, 32, 512)	2097664	up_sampling2d_1[0][0]
concatenate_1 (Concatenate)	(None, 32, 32, 1024)	0	dropout_1[0][0] conv2d_11[0][0]

Layer (type)	**Output Shape**	**Param #**	**Connected to**
conv2d_12 (Conv2D)	(None, 32, 32, 512)	4719104	concatenate_1[0][0]
conv2d_13 (Conv2D)	(None, 32, 32, 512)	2359808	conv2d_12[0][0]
up_sampling2d_2 (UpSampling2D)	(None, 64, 64, 512)	0	conv2d_13[0][0]
conv2d_14 (Conv2D)	(None, 64, 64, 256)	524544	up_sampling2d_2[0][0]
concatenate_2 (Concatenate)	(None, 64, 64, 512)	0	conv2d_6[0][0]
			conv2d_14[0][0]
conv2d_15 (Conv2D)	(None, 64, 64, 256)	1179904	concatenate_2[0][0]
conv2d_16 (Conv2D)	(None, 64, 64, 256)	590080	conv2d_15[0][0]
up_sampling2d_3 (UpSampling2D)	(None, 128, 128, 256	0	conv2d_16[0][0]
conv2d_17 (Conv2D)	(None, 128, 128, 128	131200	up_sampling2d_3[0][0]
concatenate_3 (Concatenate)	(None, 128, 128, 256	0	conv2d_4[0][0]
			conv2d_17[0][0]
conv2d_18 (Conv2D)	(None, 128, 128, 128	295040	concatenate_3[0][0]
conv2d_19 (Conv2D)	(None, 128, 128, 128	147584	conv2d_18[0][0]
up_sampling2d_4 (UpSampling2D)	(None, 256, 256, 128	0	conv2d_19[0][0]
conv2d_20 (Conv2D)	(None, 256, 256, 64)	32832	up_sampling2d_4[0][0]
concatenate_4 (Concatenate)	(None, 256, 256, 128	0	conv2d_2[0][0]
			conv2d_20[0][0]
conv2d_21 (Conv2D)	(None, 256, 256, 64)	73792	concatenate_4[0][0]
conv2d_22 (Conv2D)	(None, 256, 256, 64)	36928	conv2d_21[0][0]
conv2d_23 (Conv2D)	(None, 256, 256, 2)	1154	conv2d_22[0][0]

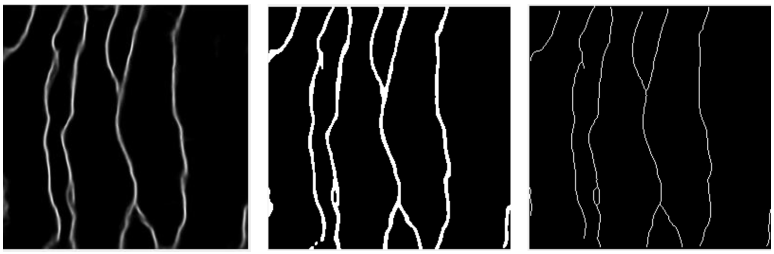
Layer (type)	Output Shape	Param #	Connected to
conv2d_24 (Conv2D)	(None, 256, 256, 1)	3	conv2d_23[0][0]

3.8.3. Postprocessing

The output images were low contrast. They were enhanced using logarithmic correction. The input gray level image undergoes pixelwise transformation according to the equation:

$$O = c * log(1 + I), \tag{8}$$

where c is a constant and l is the pixel value to be transformed. The logarithmic transformation is performed after scaling the image in the range [0, 1]. The corrected images were saved in the bitmap format. This is followed by binarization and skeletonization so that each nerve is one pixel wide only. The final segmented image is a binary image of size 256×256 pixels where each nerve pixel is represented by a value of '1' and each non-nerve pixel is represented by a value of '0'. Fig. 15 shows the postprocessing pipeline using example images.



(a) (b) (c)

Figure 15. The postprocessing pipeline. (a) the U-Net output image after logarithmic correction, (b) binary image after thresholding, (c) final segmented image after skeletonization.

3.9. Experimental Procedures for Classification

3.9.1. Preprocessing

The data was first normalized according to absolute normalization where each feature value is divided by its maximum feature value. Consequently, all feature values are in the range (0,1). This is followed by partitioning the data into subsets for training and testing. Samples are split into training and testing based on statistical similarity. This is important so that both the sets have the same distribution, and to achieve unbiased results. To achieve statistical similarity between the subsets, samples are shuffled and split into the required partition for each subset. Then, five standard statistical measures, namely, mean, median, standard deviation, fifth percentile and twenty-fifth percentile, are calculated for each feature in each set. A similarity comparison is performed to determine whether at least 3 of statistical measures for all subsets are similar within a fixed threshold. If this is not the case, then the samples are shuffled again, and the process is repeated.

3.9.2. Experimental Setup

For ANFIS, the two classifiers (C1 and C2, section 3.6) were trained for 20 epochs using a step size of 0.1 and a step increase size of 1.01. The maximum number of iterations for each classifier was empirically set to 10.

For SVM, three kernel functions were used for training: polynomial, gaussian (radial basis function) and linear.

For KNN, the discrete values of k were used from 1 to 10, and Euclidean distance was used for comparing the distances between data points.

For NB, default values were used.

3.10. Evaluation Metrics

In this section, we define the evaluation metrics used for segmentation and classification.

3.10.1. Sensitivity

Sensitivity (aka true positive rate or recall) measures the rate of actual positives over all predicted values that are actually positive. In other words, this measure gives an estimate of the proportion of actual positives that the classifier was able to recall. It is given by the formula:

$$R = \frac{TruePositives}{TruePositives + FalseNegatives}$$

For nerve segmentation, the true positives are the pixels that are correctly classified as nerve pixels, while the false negatives are the pixels that actually depicted nerves but were missed by the classifier. For neuropathy classification, recall is calculated for each class separately. For instance, for the healthy class, true positives are those images which are correctly classified as healthy, while the false negatives are the images which were misclassified as not healthy.

3.10.2. Specificity

Specificity is also called true negative rate. It is the ratio of true negatives over all negative predictions by the classifier. The higher the specificity, the better it is.

$$Specificity = \frac{TrueNegatives}{TrueNegatives + FalsePositives}$$

3.10.3. Precision

Precision is a measure that calculates the rate of actual positives out of those that are predicted by the classifier as positive. It is given by the formula:

$$P = \frac{TruePositives}{TruePositives + FalsePositives}$$

For nerve segmentation, false positives refer to those pixels that are misclassified as nerve pixels. For neuropathy classification, false positives for class ‘healthy’ are the images that are wrongly classified as healthy. A higher precision value determines a good performance.

3.10.4. Accuracy

Accuracy is the rate of total correct predictions by the classifier, given by the formula:

$$A = \frac{TruePositives + TrueNegatives}{TotalOutputs}$$

3.10.5. False Negative Rate

The false negative rate (FNR), also called the miss rate, is the rate of false negatives over the total actual positives. In nerve segmentation, it can be interpreted as the rate of nerve pixels misclassified as non-nerve. A low FNR is preferable.

$$FNR = \frac{FalseNegatives}{FalseNegatives + TruePositives}$$

3.10.6. False Positive Rate

The false positive rate (FPR) determines the ratio of false positives over all actual negatives. In nerve segmentation, it can be interpreted as the rate of predicted non-nerve pixels misclassified as nerve pixels. A low FPR is better.

$$FPR = \frac{FalsePositives}{FalsePositives + TrueNegatives}$$

3.10.7. False Detection Rate

The false detection rate (FDR) is the rate of false positives over all predicted positive. In the context of nerve segmentation, it determines the rate of pixels misclassified as nerve pixels out of all the pixels predicted as nerves. A low FDR is considered better.

$$FDR = \frac{FalsePositives}{FalsePositives + TruePositives}$$

3.10.8. Macro-F1

Macro F-measure is the harmonic mean of precision and recall, given by the formula:

$$F = 2.\frac{P*R}{P+R}$$

In multiclass problems, precision, recall and macro-F1 are calculated for each class separately.

CHAPTER 4: RESULTS AND DISCUSSION

This section provides the results from our experiments on nerve segmentation and neuropathy classification in Sections 4.1 and 4.2 respectively. Towards the end of the classification results, we also present an analysis of training time, prediction time and model size.

4.1. Nerve Segmentation

The nerve segmentation outputs obtained from U-net and ACCMetrics were compared against manual tracings by experts. Since manual tracing can sometimes waver from the nerve, a tolerance of 3 pixels was allowed for evaluation. The results for all the datasets are displayed in Table 6 in terms of average scores. In the context of nerve segmentation in corneal images, accuracy is not a very representative metric, as the majority of the pixels in the image belong to the background (non-nerve), whereas only a few belong to the foreground (nerves). Thus, more than 95% of the pixels are correctly classified because they belong to the background. Therefore, the accuracy for both methods is almost the same.

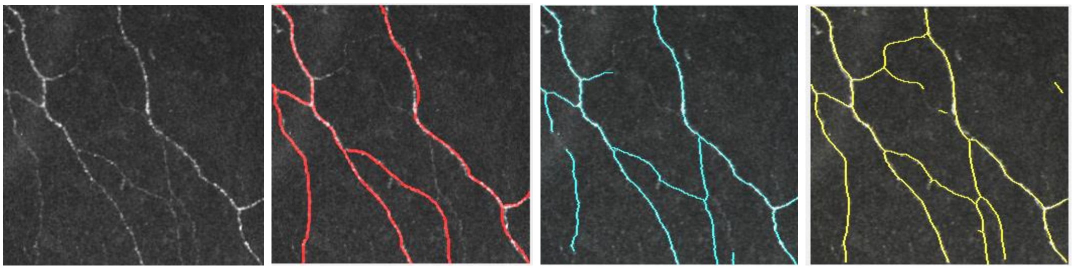
Sensitivity, however, is one of the excellent measures to evaluate nerve segmentation. As defined in Section 3.10.1, it is a measure of how many nerve pixels have been recovered. For all datasets, sensitivity of nerve pixels for U-net is significantly higher than ACCMetrics. This is a notable improvement. Similarly, the FNR for U-net is also lower than that for ACCMetrics.

Since U-net has been able to provide consistently good results (a) on the images that were captured by another ophthalmologist (DPN2, MSN1), (b) on the images from patients other than diabetes (MSN1), and (c) on the images that belong to several regions of the subbasal nerve plexus (DPN2), we can extrapolate that the U-net trained model has been able to generalize well.

Table 6. Segmentation Results

	DPN1 (n=578)		**MSN1 (n=260)**		**DPN2 (n=736)**	
	U-net	**ACCmetrics**	**U-net**	**ACCmetrics**	**U-net**	**ACCmetrics**
Accuracy	99.7%	99.4%	99.7%	99.6%	99.7%	99.7%
Sensitivity	85.1%	70.0%	80.1%	74.8%	90.2%	84.7%
Specificity	100%	100%	100%	100%	100%	100%
Precision	99.5%	99.3%	100%	99.6%	99.8%	99.7%
FNR	15.0%	30.1%	19.9%	25.2%	9.9%	15.3%
FPR	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
FDR	0.5%	0.7%	0.0%	0.4%	0.2%	0.3%

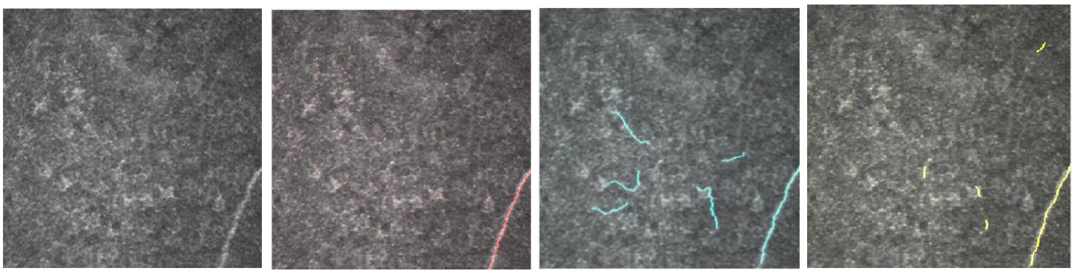
A visual example of how U-net has performed in comparison with manual tracing and ACCMetrics is shown in Figs. 16(a-d) and 17(a-d). In Fig. 16(b), two partially faded nerves have been missed by the human expert, and some portion of it was identified by ACCMetrics (Fig. 16(c)). However, U-net was able to recover all the nerves visible in the original image (Fig. 16(d)), thus increasing the recognition rate of true positives.



(a) (b) (c) (d)

Figure 16. Increase in true positives using U-net (a) Original CCM image from DPN2, (b) manual tracing (red), (c) ACCMetrics output (cyan), (d) U-net output (yellow)

In Fig. 17 (a), an example of the severe case of neuropathy is shown, where very few or no nerves are present. Due to the scarcity of nerves in the image, background texture or lens glare becomes more visible, and thus some part of the background texture has been wrongly classified as nerve pixels by ACCMetrics, as illustrated in Fig. 17(c). On the other hand, U-net significantly reduces the number of false positives (Fig. 17(d)).



(a) (b) (c) (d)

Figure 17. Reducing false positives through U-net (a) original CCM image from DPN1, (b) manual tracing (red), (c) ACCMetrics output (cyan), (d) U-net output (yellow)

Therefore, we have shown, using evaluation measures and examples, that CNN provides a better solution for nerve segmentation than ACCmetrics (RQ1).

4.2. Neuropathy Classification

In the following subsections, we present the classification results from machine learning (ANFIS, SVM, KNN, NB) and deep learning (GoogleNet transfer learning).

4.2.1. ANFIS

The results using multiclass ANFIS are displayed in Table 7. The validation accuracy of C1 and C2 are recorded for images in DPN1 only, because the training was performed on a subset of DPN1.

Table 7. ANFIS Classification Results

	DPN1		DPN2	
features	**NFL**	**NFL+ISNFL**	**NFL**	**NFL+ISNFL**
validation accuracy C1	91.6%	96.8%	N/A	N/A
validation accuracy C2	74.1%	88.9%	N/A	N/A
test accuracy	92.1%	93.7%	87.1%	77.2%
precision (0)	93.7%	97.7%	90.6%	83.6%
precision (1)	85.7%	70.6%	100.0%	58.1%
precision (2)	88.0%	95.5%	75.8%	77.8%
recall (0)	98.9%	95.6%	100.0%	95.8%
recall (1)	40.0%	80.0%	53.6%	64.3%
recall (2)	100.0%	95.5%	100.0%	56.0%
macro-f1 (0)	96.2%	96.6%	95.0%	89.3%
macro-f1 (1)	54.5%	75.0%	0.0%	61.0%
macro-f1 (2)	93.6%	95.5%	86.2%	65.1%

Observations and Discussions

- The best test accuracy provided by the ANFIS model on the DPN1 test set is 93.7%.
- The validation and test accuracy measures are close, implying that the model does not overfit.
- The use of four intra-segment nerve fiber length features improved the validation accuracies as well as the overall accuracy in DPN1.
- The use of four intra-segment nerve fiber length features also resulted in an

increase in precision for class 0 and 2, recall for class 1, and macro-f1 for all classes in DPN1.

- Using the trained model to predict DPN2, the accuracy dropped significantly. This can be expected because the image capture criteria differed from the standard established by experts at HMC. Moreover, the image quality may also be a contributing factor.

4.2.2. SVM

The classification results using three SVM kernel functions are presented in Table 8. This is followed by detailed classification results for the best performing kernel function in Table 9.

Table 8. SVM Classification Results

	DPN1	
Features	**NFL**	**NFL+ISNFL**
kernel: linear	87.4%	89.0%
kernel: polynomial	90.6%	90.6%
kernel: gaussian	88.2%	89.8%

Table 9. SVM Classification Results for the Polynomial Kernel

	DPN1		**DPN2**	
features	**NFL**	**NFL+ISNFL**	**NFL**	**NFL+ISNFL**
test accuracy	90.6%	90.6%	88.1%	89.1%
precision (0)	95.5%	96.6%	93.9%	95.7%
precision (1)	58.8%	58.8%	78.6%	81.5%
precision (2)	95.2%	90.9%	87.5%	85.2%
recall (0)	94.4%	94.4%	95.8%	93.8%
recall (1)	66.7%	66.7%	78.6%	78.6%
recall (2)	90.9%	90.9%	84.0%	92.0%
macro-f1 (0)	95.0%	95.5%	94.8%	94.7%
macro-f1 (1)	62.5%	62.5%	78.6%	80.0%
macro-f1 (2)	93.0%	90.9%	85.7%	88.5%

Observations and Discussions

- The highest accuracy achieved by SVM is 90.55%.
- The polynomial kernel function performed better than gaussian and linear functions, although the results of linear and gaussian are also close to those of polynomial.
- The addition of intra-segment features did not cause any improvement in the results for DPN1. However, the results in DPN2 are slightly better when intra-segment features were used.
- The precision and recall of the healthy and definite neuropathy classes are higher but are lower for the at-risk class.
- The results are consistent across the two datasets, implying that the model was

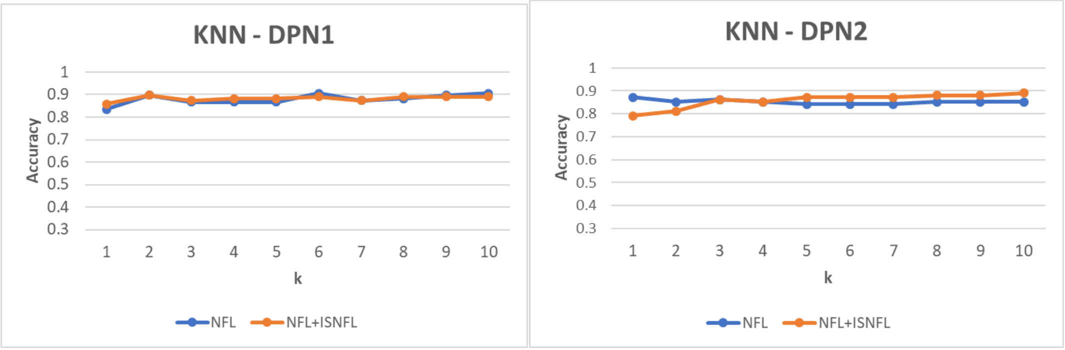
able to generalize.

4.2.3. KNN

The classification results using KNN for all values of k from 1 to 10 are presented in Table 10. The last row gives an average of the accuracy over all values of k to evaluate the potential of KNN for this problem. Figs. 18(a,b) show the effect of the values of k on accuracy when tested on the two datasets.

Table 10. KNN Classification Results

	DPN1		**DPN2**	
features	**NFL**	**NFL+ISNFL**	**NFL**	**NFL+ISNFL**
k=1	83.5%	85.8%	87.1%	79.2%
k=2	89.8%	89.8%	85.2%	81.2%
k=3	86.6%	87.4%	86.1%	86.1%
k=4	86.6%	88.2%	85.2%	85.2%
k=5	86.6%	88.2%	84.2%	87.1%
k=6	90.6%	89.0%	84.2%	87.1%
k=7	87.4%	87.4%	84.2%	87.1%
k=8	88.2%	89.0%	85.2%	88.1%
k=9	89.8%	89.0%	85.2%	88.1%
k=10	90.6%	89.0%	85.2%	89.1%
Average	88.0%	88.3%	85.2%	85.8%



(a) (b)

Figure 18. Effect of the value of k on accuracy

Observations and Discussions

- The highest test accuracy using KNN is 90.5%.
- The results are consistent across different datasets, implying that the model was able to generalize.
- Overall, the inclusion of intra-segment features showed a marginal improvement in the accuracy.
- The value of k that provided the least accuracy is observed to be 1. This result is consistent across the two datasets.

4.2.4. Naïve Bayes

The classification results using Naïve Bayes are presented in Table 11.

Table 11. Naïve Bayes Classification Results

	DPN1		DPN2	
features	**NFL**	**NFL+ISNFL**	**NFL**	**NFL+ISNFL**
test accuracy	91.3%	85.0%	89.1%	88.1%
precision (0)	95.5%	98.7%	95.8%	95.7%
precision (1)	62.5%	42.9%	79.3%	78.6%
precision (2)	95.5%	90.9%	87.5%	85.2%
recall (0)	94.4%	84.4%	95.8%	91.7%
recall (1)	66.7%	80.0%	82.1%	78.6%
recall (2)	95.5%	90.9%	84.0%	92.0%
macro-f1 (0)	95.0%	91.0%	95.8%	93.6%
macro-f1 (1)	64.5%	55.8%	80.7%	78.6%
macro-f1 (2)	95.5%	90.9%	85.7%	88.5%

Observations and Discussions

- The best accuracy obtained using NB is 91.34%.
- Precision and recall of the healthy and definite neuropathy classes are high, but the same measures for the middle class are lower.
- The results of adding the intra-segment features in NB present a paradox as compared to the results from previously mentioned classifiers. Adding the intra-segment features impacted the overall accuracy negatively. This result is consistent across the two datasets.

4.2.5. GoogleNet

The results of GoogleNet parameter tuning are shown in Fig. 19.

(a) (b)

Figure 19. GoogleNet tuning graphs. (a) tuning using the RMSProp solver, (b) tuning using the SGDM solver.

Observations and Discussions

- For both solvers, a mini batch size less than 6 reduced the accuracy. This is logical, since smaller batches are not very representative of the data and larger batch sizes are better.
- The best learning rates for RMSProp and SGDM are $2x10^{-4}$ and $1x10^{-4}$

respectively. For both solvers, a learning rate greater than $3x10^{-4}$ reduces the accuracy.

- For SGDM solver, accuracy declines with momentum. The best accuracy is at a value of 0.9 for the momentum.
- The best accuracy achieved using GoogleNet transfer learning is 85%.

4.2.6. Machine Learning and Deep Learning Results Comparison

We compared the best accuracy results obtained from all classifiers and the baseline accuracy obtained using a random classifier. We define a random classifier as a classifier that predicts the maximum class. Results from all classifiers and baseline on the two datasets are tabulated in Table 12 and illustrated graphically in Figure 20.

Table 12. All Classification Results (Test Accuracy)

	Baseline	**ANFIS**	**NB**	**SVM**	**KNN**	**GoogleNet**
DPN1	70.86%	93.70%	91.34%	91.34%	90.55%	85.83%
DPN2	47.52%	87.12%	89.10%	89.10%	89.11%	81.89%

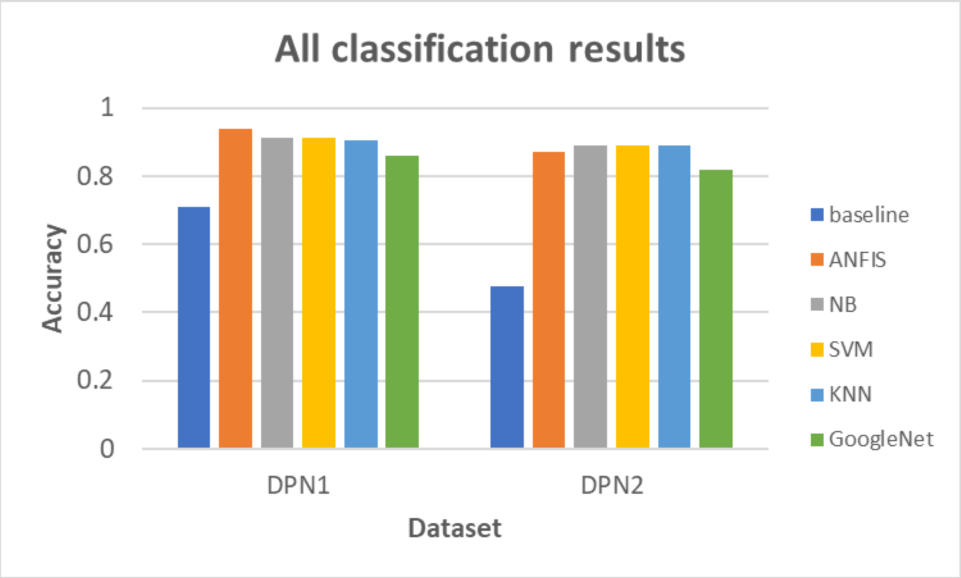


Figure 20. Classification Results from All Techniques

Thus, we arrive at the following conclusions regarding our research questions. The highest classification accuracy was achieved by ANFIS at 93% and the lowest performance was observed by GoogleNet. The precision of the middle class (at-risk neuropathy) was the most difficult for classifiers to understand. The use of intra-segment features resulted in marginal improvement in the overall accuracy (RQ2). After an evaluation of five learners, we can observe that ANFIS has shown the highest accuracy (RQ3). Moreover, for this problem, simple machine learning proved to be a better approach than CNNs (RQ4).

The results of this study cannot be compared with the results from the two similar studies [37], [56] found in the literature due to several reasons. First of all, the protocol for setting the ground truth was different; previous studies have classified based on the neuropathy classification done using the existing methods for detecting neuropathy. On the other hand, our ground truth classification is based on the visual information from the images themselves. Secondly, the image capture protocol is different. Our machine learning models are trained on images captured strictly from the central region of the cornea, whereas the images in previous studies included images from the inferior whorl as well. Thirdly, our images are strictly nonoverlapping; the images in the previous studies do not imply this condition. Although their results cannot be compared, we have used their dataset and handpicked some images that satisfy our image criteria for classification. In this way we attempted to provide a fair comparison on different datasets, instead of comparing our results with their published results.

4.2.7. Training and Prediction Time Analysis

The complete training and prediction times for each technique were measured on DPN1. The approximate elapsed times on a CPU are displayed in Table 13. It can be seen that GoogleNet requires the most time for training as well as prediction when compared to the others. This is because deep neural networks in general require a lot of time to learn from the data, and build complex models to predict on. On the other hand, other learners do not have intensive computation during the prediction.

*Table 13. Training and Prediction Times**

	ANFIS	**SVM**	**NB**	**KNN**	**GoogleNet**
Training time	78s	1.8s	0.198s	0.095s	10 minutes
Testing time	0.005s	0.065s	0.048s	0.051s	7.1s

**Training and prediction times are calculated for the whole test set*

4.2.8. Model Size Analysis

Although disk size is a trivial issue in the current time when terabytes of storage and gigabytes of disk space is available for processing on computers, an analysis of the size of the trained model is generally important when it is to be deployed on the relevant hardware device which has limited capacity. In this problem, the deployment of the model on the corneal confocal microscope can help the doctors to achieve at an automated classification result immediately after image capture, or even while scanning the cornea. Although this feature is not implemented in the current study, we present an analysis of the model sizes of all the classifiers used (Table 14). Apparently, the GoogleNet model requires the most space as compared to machine learning approaches, however, it must be noted that we used unsegmented images for GoogleNet. On the other hand, the machine learning approaches required segmentation through a CNN before classification, which is again a deep convolutional neural network requiring more than 30MB. Therefore, the approaches presented here are more suited for an offline processing of images.

Table 14. Model Sizes

	ANFIS	**SVM**	**NB**	**KNN**	**GoogleNet**
Model Size	70KB	21KB	23KB	53KB	22MB

4.2.9. Statistical Significance

One of the methods for performing statistical agreement analysis is by calculating the interrater reliability using the kappa statistic. As stated by Landis & Koch [61], values of Kappa less than zero show poor agreement, and the strength of agreement increases as the value of kappa reaches 1.0. A substantial level of agreement is shown by values ranging between 0.61 and 0.80, and values ranging from 0.81 to 1 show almost perfect agreement.

We calculated the kappa score between ground truth and the output from each of the machine learning models and their corresponding p-values. Results show that the outputs from all models are statistically significant (p<0.0001). The kappa value shows an almost perfect agreement of the ground truth with the ANFIS output and substantial agreement with SVM, KNN and NB. These results were computed using the IBM SPSS software.

Table 15. Kappa Statistics and p-values

	GT * ANFIS	**GT * SVM**	**GT * NB**	**GT * KNN**
kappa	0.864	0.796	0.709	0.776
p-value	0.000	0.000	0.000	0.000

CHAPTER 5: CONCLUSION

This thesis research focuses on evaluating the potential of various machine learning algorithms for neuropathy classification of CCM images. In addition, we investigate whether deep neural networks are suitable for this problem. Moreover, we compare a state-of-the-art approach for nerve segmentation of CCM images with a recently proposed approach in the literature using CNN.

The experiments for nerve segmentation were performed on three datasets using ACCmetrics and a U-net based CNN. The results using CNN showed a significantly high sensitivity and precision and low FNR and FDR when compared with the results from ACCMetrics. These results were consistent across the three datasets.

We conducted experiments for classification on images from two datasets. Overall, we observed that (1) the results greatly improved over baseline (random classifier), and that (2) machine learning algorithms provided better results than deep learning. However, we cannot yet conclude that machine learning algorithms outperform the deep learning approach, as further experiments on more data must be conducted to establish such a claim. Furthermore, the use of intra-segment features showed an improvement in the classification accuracy for ANFIS, a decline in the classification accuracy for NB, and no significant effect for other learners. We believe that this hypothesis will hold true for other learners as well once the medical experts add further dimensions to their definition of neuropathy in the images.

In our experiments, we achieved the best accuracy at 93.7% using the ANFIS classifier. We further observed that, in all experiments, the classification of the middle class (at-risk neuropathy) was the most difficult for all classifiers to perform resulting in a consistently lower recall for class 1 (at-risk). In general, most of the misclassifications occurred between class 0 and 1 (healthy and at-risk), and class 1 and 2 (at-risk and definite). Misclassifications between class 0 and 2 (healthy and definite) were negligible. Additionally, our results are statistically significant ($p<0.0001$) and the kappa scores indicate a substantial agreement between the ground truth and predicted outputs.

5.1. Future Research Directions

There are many directions that this research could take in the future contributing to the benefit of the general community and adding useful insights on neuropathy for the scientific community. We list some of them below:

1. In this study, we focused on image-based neuropathy classification. We plan to extend this research to patient-based classification through machine learning and deep learning techniques. The goal will be to test the effectiveness of different approaches in risk-stratifying patients based on their CCM result and creating a heat map visualization of areas depicting neuropathy. This involves analyzing a vast amount of CCM images from different conditions, generating ground truth data for nerves and Langerhans cells to facilitate accurate analysis.
2. Currently, we have considered images from diabetic patients only, since the appearance of diabetic neuropathy in the human cornea is a well-established fact and reported by many studies. What remains to be explored is whether different diseases affect the cornea differently in terms of neurodegeneration. For this purpose, we plan to apply artificial intelligence techniques to understand what nerve loss looks like in various disease conditions and its relationship to clinical outcomes.
3. The experiments and results conducted in this study apply only to CCM images captured from the central region of the cornea. We intend to include the use of

inferior whorl images to define the relationship between percent change at baseline versus follow-up and quantify the risk of disease progression.

4. As the study will proceed, more factors will arise that define the neuropathy condition. Our future studies will determine enhanced features from CCM images that are essential for identifying the neuropathy condition. We will also attempt to quantify nerve clustering, tissue reflection, presence of Langerhan cells in CCM images.
5. In this research we relied on identifying neuropathy through images and neglected the patient history. A robust and reliable system in the future will include information from the images as well as patient's clinical history. We will also determine which features from the patient history are of relevance.
6. Lastly, a very important aspect of the study would be to investigate and seek to establish the relationship between the different modalities that reflect the neuropathy condition of a patient with respect to a disease. These modalities include CCM images, corpus callosum in the brain MRI scans and the cross-sectional retina captured in OCT scans.

5.2. Related Publications

[1] T. Salahuddin and U. Qidwai, “Neuro-Fuzzy Classifier for Corneal Nerve Images,” in IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES), Malaysia, 2018, pp. 131–136.

[2] T. Salahuddin and U. Qidwai, “Classification of Corneal Nerve Images Using Machine Learning Techniques,” in International Journal of Integrated Engineering. (Accepted: April 2019)

[3] T. Salahuddin, S. AlMaadeed, I. N. Petropolous, R. A. Malik, S. K. Ilyas, & U. Qidwai, “Smart Neuropathy Detection using Machine Intelligence: Filling the Void Between Clinical Practice and Early Diagnosis,” in IEEE World Conference on Smart Trends in Systems, Security and Sustainability, London, UK. (Accepted: May 2019)

REFERENCES

[1] C. Quattrini *et al.*, "Surrogate markers of small fiber damage in human diabetic neuropathy," *Diabetes*, 2007.

[2] M. A. Dabbah, J. Graham, R. Malik, and N. Efron, "Detecting and Analyzing Linear Structures in Biomedical Images: A Case Study Using Corneal Nerve Fibers," in *Medical Image Processing*, 2011.

[3] N. Efron, "Corneal Confocal Microscopy Is Emerging as a Powerful Diagnostic Tool for Assessing Systemic Neurologic Disease," *Invest. Ophthalmol. Vis. Sci.*, vol. 58, no. 9, pp. 3682–3682, 2017.

[4] P. Hossain, A. Sachdev, and R. A. Malik, "Early detection of diabetic peripheral neuropathy with corneal confocal microscopy," *Lancet*. 2005.

[5] M. Tavakoli, P. Begum, J. Mclaughlin, and R. A. Malik, "Corneal confocal microscopy for the diagnosis of diabetic autonomic neuropathy," *Muscle and Nerve*, 2015.

[6] M. Brines *et al.*, "Corneal nerve fiber size adds utility to the diagnosis and assessment of therapeutic response in patients with small fiber neuropathy," *Sci. Rep.*, 2018.

[7] I. N. Petropoulos *et al.*, "Rapid automated diagnosis of diabetic peripheral neuropathy with in vivo corneal confocal microscopy," *Investig. Ophthalmol. Vis. Sci.*, 2014.

[8] N. Lagali *et al.*, *Laser-scanning in vivo confocal microscopy of the cornea: Imaging and analysis methods for preclinical and clinical applications*. 2013.

[9] A. Khan *et al.*, "Corneal confocal microscopy detects corneal nerve damage in patients admitted with acute ischemic stroke," *Stroke*, 2017.

[10] P. J. Podgorny, O. Suchowersky, K. G. Romanchuk, and T. E. Feasby, "Evidence

for small fiber neuropathy in early Parkinson's disease," *Park. Relat. Disord.*, 2016.

[11] L. Kass-Iliyya *et al.*, "Small fiber neuropathy in Parkinson's disease: A clinical, pathological and corneal confocal microscopy study," *Park. Relat. Disord.*, 2015.

[12] M. Ferdousi *et al.*, "Corneal confocal microscopy detects small fibre neuropathy in patients with upper gastrointestinal cancer and nerve regeneration in chemotherapy induced peripheral neuropathy," *PLoS One*, 2015.

[13] H. I. Kemp *et al.*, "Use of corneal confocal microscopy to evaluate small nerve fibers in patients with human immunodeficiency virus," *JAMA Ophthalmol.*, 2017.

[14] N. L.-O. T. XXIX and undefined 2019, "Need for technologies in advanced corneal research, diagnosis, and transplantation," *spiedigitallibrary.org*.

[15] M. A. Dabbah, J. Graham, I. N. Petropoulos, M. Tavakoli, and R. A. Malik, "Automatic analysis of diabetic peripheral neuropathy using multi-scale quantitative morphology of nerve fibres in corneal confocal microscopy imaging," *Med. Image Anal.*, 2011.

[16] M. A. Dabbah, J. Graham, I. Petropoulos, M. Tavakoli, and R. A. Malik, "Dual-model automatic detection of nerve-fibres in corneal confocal microscopy images," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2010.

[17] X. Chen, J. Graham, M. A. Dabbah, I. N. Petropoulos, M. Tavakoli, and R. A. Malik, "An automatic tool for quantification of nerve fibers in corneal confocal microscopy images," in *IEEE Transactions on Biomedical Engineering*, 2017.

[18] J. S. R. Jang, "ANFIS: Adaptive-Network-Based Fuzzy Inference System,"

IEEE Trans. Syst. Man Cybern., 1993.

[19] N. Talpur, M. N. M. Salleh, and K. Hussain, "An investigation of membership functions on performance of ANFIS for solving classification problems," in *IOP Conference Series: Materials Science and Engineering*, 2017.

[20] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, 1995.

[21] I. Rish, "An empirical study of the naive Bayes classifier," in *International Joint Conferences on Artificial Intelligence 2001 Workshop on Empirical Methods in Artificial Intelligence*, 2001.

[22] T. M. Cover and P. E. Hart, "Nearest Neighbor Pattern Classification," *IEEE Trans. Inf. Theory*, 1967.

[23] S. A. Dudani, "The Distance-Weighted k-Nearest-Neighbor Rule," *IEEE Trans. Syst. Man Cybern.*, 1976.

[24] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015.

[25] A. Ruggeri, F. Scarpa, and E. Grisan, "Analysis of corneal images for the recognition of nerve structures," in *Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE*, 2006, vol. 4739–4742.

[26] A. Grisan, E., Pesce, A., Giani, A., Foracchia, M., & Ruggeri, "A new tracking system for the robust extraction of retinal vessel structure," in *Engineering in Medicine and Biology Society, 2004. IEMBS'04. 26th Annual International Conference of the IEEE*, p. Vol. 1, pp. 1620–1623.

[27] F. Scarpa, E. Grisan, and A. Ruggeri, "Automatic recognition of corneal nerve

structures in images from confocal microscopy," *Investig. Ophthalmol. Vis. Sci.*, 2008.

[28] E. Poletti and A. Ruggeri, "Automatic nerve tracking in confocal images of corneal subbasal epithelium," in *Computer-Based Medical Systems (CBMS), 2013 IEEE 26th International Symposium,* 2013, pp. 119–124.

[29] R. N. Dixon and C. J. Taylor, "Automated Asbestos Fiber Counting," in *1979 Inst. Physics Conference,* 1979.

[30] M. A. Dabbah, J. Graham, M. Tavakoli, Y. Petropoulos, and R. A. Malik, "Nerve fibre extraction in confocal corneal microscopy images for human diabetic neuropathy detection using gabor filters," *Med. Image Underst. Anal.*, vol. 254–258, 2009.

[31] S. Al-Fahdawi *et al.*, "A fully automatic nerve segmentation and morphometric parameter quantification system for early diagnosis of diabetic neuropathy in corneal images," *Comput. Methods Programs Biomed.*, 2016.

[32] P. Guimaraes, J. Wigdahl, E. Poletti, and A. Ruggeri, "A fully-automatic fast segmentation of the sub-basal layer nerves in corneal images," in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC 2014,* 2014.

[33] P. Rani, N. Priyadarshini, E. R. Rajkumar, and K. Rajaman, "Retinal vessel segmentation under pathological conditions using supervised machine learning," in *International Conference on Systems in Medicine and Biology (ICSMB) IEEE,* 2016.

[34] T. Jerman, F. Pernuš, B. Likar, and Ž. Špiclin., "Enhancement of vascular structures in 3D and 2D angiographic images," *IEEE Trans. Med. Imaging,* 2016.

[35] H. Fu, Y. Xu, D. W. K. Wong, and J. Liu, “Retinal vessel segmentation via deep learning network and fully-connected conditional random fields,” in *Proceedings - International Symposium on Biomedical Imaging*, 2016.

[36] S. Xie and Z. Tu, “Holistically-Nested Edge Detection,” *Int. J. Comput. Vis.*, 2017.

[37] A. Colonna, F. Scarpa, and A. Ruggeri, “Segmentation of Corneal Nerves Using a U-Net-Based Convolutional Neural Network,” 2018, pp. 185–192.

[38] H. Dong, G. Yang, F. Liu, Y. Mo, and Y. Guo, “Automatic Brain Tumor Detection and Segmentation Using U-Net Based Fully Convolutional Networks,” 2017, pp. 506–517.

[39] P. Costa *et al.*, “Towards adversarial retinal image synthesis,” *arXiv Prepr.*, 2017.

[40] J. Son, S. J. Park, and K.-H. Jung, “Retinal Vessel Segmentation in Fundoscopic Images with Generative Adversarial Networks,” Jun. 2017.

[41] K.-K. Maninis, J. Pont-Tuset, P. Arbeláez, and L. Van Gool, “Deep Retinal Image Understanding,” 2016, pp. 140–148.

[42] C. Szegedy *et al.*, “Going deeper with convolutions,” *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, 2015.

[43] A. Krizhevsky, I. Sutskever, and G. H. Neural, “Imagenet classification with deep convolutional neural networks,” *Adv. Neural Inf. Process. Syst.*, 2012.

[44] V. Gulshan *et al.*, “Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs,” *Jama*, 2016.

[45] J. Y. Choi, T. K. Yoo, J. G. Seo, J. Kwak, T. T. Um, and T. H. Rim, “Multi-categorical deep learning neural network to classify retinal images: A pilot study employing small database,” *PLoS One*, vol. 12, no. 11, p. e0187336, Nov. 2017.

[46] C. Lam, D. Yi, M. Guo, and T. Lindsey, “Automated Detection of Diabetic Retinopathy using Deep Learning,” in *AMIA Summits on Translational Science Proceedings 2017*, 2018.

[47] P. Burlina, D. E. Freund, N. Joshi, Y. Wolfson, and Neil M. Bressler, “Detection of age-related macular degeneration via deep learning,” in *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*, 2016.

[48] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, “Overfeat: Integrated recognition, localization and detection using convolutional networks,” *arxiv.org*, 2013.

[49] P. M. Burlina, N. Joshi, M. Pekala, K. D. Pacheco, D. E. Freund, and N. M. Bressler, “Automated grading of age-related macular degeneration from color fundus images using deep convolutional neural networks,” *JAMA Ophthalmol.*, vol. 135, pp. 1170–1176, 2017.

[50] R. Poplin *et al.*, “Prediction of cardiovascular risk factors from retinal fundus photographs via deep learning,” *Nat. Biomed. Eng.*, 2018.

[51] D. S. W. Ting *et al.*, “Development and validation of a deep learning system for diabetic retinopathy and related eye diseases using retinal images from multiethnic populations,” *Jama*, vol. 318, pp. 2211–2223, 2017.

[52] M. D. Abràmoff *et al.*, “Improved automated detection of diabetic retinopathy on a publicly available dataset through integration of deep learning,” *Invest. Ophthalmol. Vis. Sci.*, vol. 57, pp. 5200–5206, 2016.

[53] H. Takahashi, H. Tampo, Y. Arai, Y. Inoue, and H. Kawashima, “Applying artificial intelligence to disease staging: Deep learning for improved staging of diabetic retinopathy,” *PLoS One*, vol. 12, no. 6, p. e0179790, Jun. 2017.

[54] J. H. Tan *et al.*, “Age-related Macular Degeneration detection using deep

convolutional neural network," *Futur. Gener. Comput. Syst.*, vol. 87, pp. 127–135, 2018.

[55] R. Gargeya and T. Leng, "Automated identification of diabetic retinopathy using deep learning," *Ophthalmology*, vol. 124, pp. 962–969, 2017.

[56] S. F. Silva *et al.*, "Diabetic peripheral neuropathy assessment through texture based analysis of corneal nerve images," *J. Phys. Conf. Ser.*, vol. 616, p. 012002, May 2015.

[57] I. Otel *et al.*, "Diabetic peripheral neuropathy assessment through corneal nerve morphometry," in *Bioengineering (ENBENG), 2013 IEEE 3rd Portuguese Meeting*, 2013.

[58] I. N. Petropoulos *et al.*, "Corneal nerve loss detected with corneal confocal microscopy is symmetrical and related to the severity of diabetic polyneuropathy," *Diabetes Care*, vol. 36, pp. 3646–3651, 2013.

[59] G. Varoquaux, L. Buitinck, G. Louppe, O. Grisel, F. Pedregosa, and M. Andreas, "Scikit-learn: Machine learning without learning the machinery," *GetMobile Mob. Comput. Commun.*, vol. 12, pp. 2825–2830, 2011.

[60] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16),* 2016, pp. 265–283.

[61] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *Biometrics*, vol. 33, no. 1, pp. 159–174, 1977.