RETINAL FUNDUS IMAGE PROCESSING AND ENSEMBLE LEARNING: OPTIC DISC AND OPTIC CUP DETECTION

by

Shima Mohammadali Pishnamaz

Bachelor of Engineering, Ryerson University, Toronto, Canada, 2015

A thesis

presented to Ryerson University in partial fulfillment of the requirements for the degree of Master of Applied Science

in the Program of

Electrical & Computer Engineering

Toronto, Ontario, Canada, 2018

© Shima Mohammadali Pishnamaz, 2018

AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A THESIS

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners. I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research. I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research. I understand that my thesis may be made electronically available to the public.

Shima Mohammadali Pishnamaz

Retinal Fundus Image Processing and Ensemble Learning: Optic Disc and Optic Cup Detection Master of Applied Science 2018 Shima Mohammadali Pishnamaz Electrical & Computer Engineering Ryerson University

Abstract

Ophthalmologists have widely used retinal fundus imaging systems to examine the health of the optic nerve, vitreous, macula, retina and their blood vessels. Many critical diseases, such as glaucoma and diabetic retinopathy, can be diagnosed by analyzing retinal fundus images. Retinal image-based glaucoma detection is a comprehensive diagnostic approach that examines the head cup-to-disc ratio (CDR) as an important indicator for detecting the presence and the extent of glaucoma in a patient. The accurate segmentations of the optic disc (OD) and optic cup (OC) are critical for the calculation of CDR. Machine learning based algorithms can be very helpful to efficiently exploit the vast amounts of retinal fundus data.

In this thesis project, the main goal is to develop image processing and machine learning algorithms to automatically detect OD and OC from fundus images. This goal has been achieved by developing and applying several image enhancement techniques. First, an algorithm is proposed and tested on several fundus images to detect OD. The proposed algorithm is based on a combination of Contrast Limited Adaptive Histogram Equalization (CLAHE), Alternating Sequential Filters (ASF), thresholding, and Circular Hough Transform (CHT) methods. The results section highlights that the proposed algorithm is highly efficient in segmentation of OD from other

parts of the fundus image. Several classification and modeling methods are studied in order to classify detected OD into OC and non-OC regions. In this thesis project three main ensemble modeling algorithms are studied to segment OC. The studied ensemble models are Random Forest, Gradient Boosting Machines (GBM), and Extreme Gradient Boosting Machines (XGBoost). The comparison between these models shows that they have more accurate results than conventional classification methods such as Logistic Regression (LR) or Support Vector Machines (SVM). This study shows that XGBoost is the fastest and most accurate approach to segment optic cup within the optic disc region.

Acknowledgements

I would like to thank my supervisor, Dr. Kaamran Raahemifar for his guidance and support during my studies. I feel privileged to have the opportunity of working with him. I would like to thank the members of my master's defense committee, Dr. Dafna Sussman and Dr. Sri Krishnan. I would also like to thank my collaborators from the University of Waterloo, Dr. Vasudevan Lakshminarayanan and Dr. Ahmed Almazroa who provided me with constructive guidance throughout my research studies.

Last but not least, I would like to express my gratitude to my family. I am grateful to my parents who supported and encouraged me in all the stages of my life. I also want to thank my husband, Pooya, for always supporting and loving me unconditionally. This accomplishment would not have been possible without him.

Table of Contents

Table of Contents
List of Tables ix
List of Figures x
List of Abbreviations
Chapter 1: Introduction
1.1. Motivation
1.2. Goals and Specific Aims
1.3. Main Contributions of the Thesis
1.4. Overview of the Thesis
Chapter 2: Background
2.1. Retina
2.2. Retinal Imaging
2.2.1. Diabetic Retinopathy
2.2.2. Glaucoma
2.3. Retinal Fundus Imaging

2.4. O _I	ptic Disc and Optic Cup Detection	12
Chapter 3:]	Retinal Fundus Image Processing	17
3.1. Oj	ptic Disc Detection	17
3.1.1.	RGB Channel Extraction, Normalization and CLAHE	19
3.1.2.	Morphological Operators: Alternating Sequential Filtering	20
3.1.3.	Thresholding	21
3.1.4.	Circular Hough Transform	21
3.1.5.	Gabor Filter	22
3.1.6.	Feature Extraction	23
3.2. Cl	assification and Optic Cup Detection	24
3.2.1.	Conventional Methods	24
3.2.2.	Ensemble Methods	27
3.2.3.	Random Forest	28
3.2.4.	Gradient Boosting Machine	32
3.2.5.	Comparison between Random Forest and Gradient Boosting Machine	35
3.2.6.	Extreme Gradient Boosting (XGBoost)	37
3.2.7.	Classification Method Selection	42
3.3. Di	stributed Systems	43
3.3.1.	Different Types of Distributed Systems	44
3.3.2.	Apache Spark Overview	48

Chapter 4: Results and Discussions 50		
4.1.	Optic Disc Detection Results	50
4.2.	Optic Cup detection Results	61
Chapter	5: Conclusions and Future Work	68
5.1.	Conclusions	68
5.2.	Future Work	69
Referen	ces:	70

List of Tables

Table 4.1. Evaluation of optic disc detection algorithm 60
Table 4.2. Comparison of the proposed method and other studies 61
Table 4.3. Evaluation of Logistic Regression Algorithm 62
Table 4.4. Evaluation of Support Vector Machines 63
Table 4.5. Evaluation of Gradient Boosting Machines 63
Table 4.6. Evaluation of Random Forest Algorithm 64
Table 4.7. Results from XGBoost with different hyperparameters 65
Table 4.8. Evaluation of XGBoost Algorithm after Tuning
Table 4.9. Evaluation Summaries
Table 4.10. Comparison Between Different Algorithms in Optic Cup Detection

List of Figures

Figure 2.1. Eye Anatomy
Figure 2.2. The Cup-to-Disc Ratio in (a) normal, (b) Glaucoma retina13
Figure 3.1. The flowchart of the proposed method for segmentation of optic disc in retinal color
fundus image
Figure 3.2. Decision Tree
Figure 3.3. (a) Random Forest vs. (b) Gradient Boosting Machine
Figure 3.4. OC Classification Algorithms
Figure 3.5. Master-Slave Replication Strategy45
Figure 3.6. CAP Theorem: Consistency, Availability, and Partition Tolerance Overlaps
Figure 3.7. Hadoop Distributed File System Architecture
Figure 3.8. Apache Spark Cluster Overview
Figure 4.1. Six sample retinal fundus images from MESSIDOR database
Figure 4.2. Red Channel
Figure 4.3. Green Channel53
Figure 4.4. Blue Channel
Figure 4.5. Red Channel after CLAHE55
Figure 4.6. Red Channel after CLAHE, ASF, and Thresholding56
Figure 4.7. Red Channel after CLAHE, ASF, Thresholding, and CHT57

Figure 4.8.	Comparison betw	veen optic discs	detected by th	e algorithm vs	the experts	
Figure 4.9.	Comparison betw	veen optic discs	detected by th	e algorithm vs	the experts	59

List of Abbreviations

ACM	Active Contour Model
AHE	Adaptive Histogram Equalization
ANN	Artificial Neural Network
ASF	Alternative Sequential Filters
CDR	Cup to Disc Ratio
CHT	Circular Hough Transform
CLAHE	Contrast Limited Adaptive Histogram Equalization
FCM	Fuzzy C-Means
GBM	Gradient Boosting Machines
LBP	Local Binary Patterns
LDA	Linear Discriminant Analysis
LR	Logistic Regression
NPDR	Non-Proliferative Diabetic Retinopathy
OC	Optic Cup
OCT	Optical Coherence Tomography
OD	Optic Disc
PDR	Proliferative Diabetic Retinopathy
RGB	Red-Green-Blue
ROI	Region of Interest
PSO	Particle Swarm Optimization
SE	Structural Element
SNR	Signal to Noise Ratio
SVM	Support Vector Machines
XGBoost	Extreme Gradient Boosting Machines

Chapter 1

Introduction

In this chapter, the motivations of this thesis study are first introduced. Then the goals and the specific aims are listed and briefly explained. The contributions of this research are also described and, finally, an overview of the thesis is presented.

1.1. Motivation

Retinal fundus imaging is a highly specialized form of medical imaging. These images are an important means to document the health of the optic nerve, vitreous, macula, retina and its blood vessels [1]. Many eye diseases as well as systemic diseases manifest themselves in the retina [2]. Retinal fundus images are used for diagnosing diseases such as age-related macular degeneration, diabetes, and glaucoma [1], [2]. Nowadays, huge amount of retinal data is available to be leveraged for added clinical knowledge, however currently this valuable information is not being properly used. There are important detectable signs on retinal images which are comprehensive for identification of glaucoma, therefore retinal image processing can help extensively in diagnosis and monitoring of glaucoma. One of these signs is the ratio of optic cup to optic disc, known as cup-to-disc ratio (CDR). Accurate segmentation of optic disc and optic cup from retinal images is

an initial step towards obtaining and calculating CDR and eventually diagnosis and monitoring of glaucoma [1].

1.2. Goals and Specific Aims

The main goal of this study is to develop an algorithm with the capacities to automatically and accurately detect OD and OC from fundus images. Such detection method can help ophthalmologists diagnose glaucoma. To achieve this goal, several image processing algorithms are studied and applied to the fundus images. For the purpose of OC detection, the ensemble modelling methods are studied and applied, including Random Forest, Gradient Boosting Machines (GBM), and Extreme Gradient Boosting Machines (XGBoost).

The specific aims of this research are:

- Development and implementation of an image processing algorithm that automatically segments OD from fundus images. This step includes implementation of image enhancement algorithms as well as segmentation of RGB images.
- 2- Introducing a set of features to define meaningful predictors from the OD region.
- 3- Implementation of predictive models to classify the image pixels of the OD region into OC and non-OC pixels.
- 4- Evaluating ensemble modeling methods, such as Random Forest, GBM and XGBoost, in classifying OD pixels into OC and non-OC pixels in OD region and comparing their results with conventional classification methods such as Logistic Regression and Support Vector Machines (SVM).

1.3. Main Contributions of the Thesis

The main contributions of this work can be summarized as in the following:

- 1- Development of an algorithm to accurately detect OD in retinal fundus images,
- 2- Investigations over the Bagging and Boosting ensemble models to classify OC pixels,
- 3- Development of an accurate and fast ensemble model based on XGBoost to efficiently classify OC pixels,
- 4- Evaluation of the results from ensembles with the conventional classification methods such as Logistic Regression and SVM.

1.4. Overview of the Thesis

Chapter 2 includes the background of retinal fundus imaging. The methods from the other studies are also summarized in this chapter.

The proposed methods for OD and OC segmentation are described in more details in Chapter 3.

In Chapter 4, the results of the proposed methods are provided and evaluated. This chapter also includes discussions of results from the proposed methods.

In Chapter 5, the summary of this study and future works are provided.

Chapter 2

Background

In this chapter, first retina and retinal imaging are briefly explained. Then retinal fundus imaging and the methods used to detect OD and OC are reviewed.

2.1. Retina

Retina is a thin layer of tissue on the inside of the eye. It is located near the optic nerve. The purpose of retina is to receive the light that the lens focuses, convert it into neural signals, and send these signals on to the brain for visual recognition [2]. Figure 2.1 shows eye anatomy including retina, optic disc and optic cup [3].



Figure 2.1. Eye Anatomy [3]

The functionality of retina in the eye is the same as a sensor in a digital camera. Chemical reactions happen inside the photoreceptors of the retina every time light hits the eye. After the occurrence of the chemical reactions, electrical signals will be transmitted to the brain through nerve cells. Rods and cones are two main types of cells in the retina. Cones enable color vision and require more light in the environment. Rods, on the other hand, do not enable color vision and are responsible for seeing in darker environments. The most number of cones are found in the central and most functional part of retina also known as Macula. Any kind of damage to Macula causes dark and distorted vision [4].

2.2. Retinal Imaging

Retina deals with the light coming from outside of the eye; therefore, the ocular parts of it are transparent and available for image creation through proper imaging approaches [5]. Retina is a highly metabolic and active structure in the eye which has a double blood supply, this characteristic makes the retina to be a suitable eye structure for the application of non-invasive imaging. Normally high resolution imaging approaches are leveraged for taking images from inside of the eye. The images taken from the eye can help ophthalmologists and optometrists to identify signs of eye disorders and eye damages. A number of diseases that manifest in the aye are diabetes, high blood pressure, retinal tearing, retinal detachment and glaucoma [6]–[9]. Early detection of eye disease is very critical as it can prevent one from losing vision.

Today, ophthalmologists have a vast array of imaging tools to choose from. There has been extensive research and development done in different areas of image processing for the purpose of diagnosis, monitoring and treatment of severe eye complexities. This extensive work has resulted in many newly developed state of the art techniques that are made available for clinical use. Examples of such techniques include Fundus Photography, Optical Coherence Tomography (OCT), Optical Coherence Tomography Angiography (OCT-Angiopathy) and Fluorescein Angiopathy. These techniques are used in eye disease detection and in monitoring the progression of the diseases over time [10], [11].

Fundus imaging is an image preessing technique which is normally used in clinical trials for diagnosing retinal diseases [12]. There are three forms of fundus photography techniques; colour fundus photography, stereo fundus photography and red-free photography. Eye complications can be detected using fundus images. As an example, diabetic retina could be diagnosed through color fundus images showing specific features such as intra-retinal haemorrhages, intra-retinal microvascular abnormalities, microaneurysms, cotton-wool spots and venous beadings [2], [13]–[15]. Colour fundus photographs are also used in analysis of eye in order to diagnose and monitor hard exudates, photocoagulation burns and retinal alterations [9].

Another imaging technique is known as Fundus Autoflourescence which detects lipofuscin pigments in retinal epithelium. Lipofuscin pigments do appear in abnormal outer retina which could be a sign of aging cells [7]. Laser ophthalmoscopy is yet another imaging technique which uses infrared light to illuminate the retina and is used in imaging in cloudy environments in small pupils. This technique can be leveraged for detection of retinal hyperpigmentation, hard exudates and subretinal fluid in bloodstream. It is also used in diagnosing cystoid macular oedema. Laser ophthalmoscopy enables localization of the cystic features in patients [7].

The gold standard method for in-vivo monitoring of retinal vasculature is Flourescein Angiopathy. This technique has been used over the past thirty years to monitor chorioretinal diseases [7]. It enables the identification of fluorescein leakage source for macular laser treatment. It is also used in analysing response to treatment process.

OCT is another widely used tools in clinical and research practices over the past two decades [7], [10]. It enables high-resolution and non-invasive imaging that can assess retinal thickness and morphology, and choroidal morphology in patients. There are three types of OCT: time domain, spectral-domain, and swept-source. Spectral domain devices are currently the most commonly used commercial OCT devices [10], [16].

Another relatively new imaging modality is known as Optical Coherence Tomography Angiopathy (OCT-angiopathy) which is used in three dimensional monitoring of chosoidal eye vessles without using any contrast dye. Movement of erythrocytes between individual scans causes decorrelation signals which are used in order to generate the image. Specific techniques are then applied in order to clean the images and reduce signal-to-noise ratio (SNR) [7], [10].

A method known as Hyperspectrual imaging utilizes the difference in spectural properties of haemoglobin. Difference in arterio-venous oxygen saturation can be an indicator in identifying state of a certain disease.

As briefly explained in this section, there has been extensive research and techniques developed for the diagnosis and monitoring of eye diseases over the past few years. This thesis study focuses on digital fundus imaging technique.

Multiple studies have linked geometric retinal vascular parameters with vascular diseases including ischaemic heart disease, hypertension, stroke and diabetes [2], [4], [17].

In the following, a brief description of these disorders are summarized:

<u>Age-related Macular Degeneration</u>: Macular degeneration is a common eye condition and a leading cause of central vision loss among people of age 50 and older. It is usually signified by fluid leaking or bleeding in the back of the eye [1].

<u>**Cancer</u>**: Melanoma is one type of cancer that can grow without any major noticeable sign in the retina. Mealnoma usually occurs on the skin, however in some rare cases it can also occur in other parts of the body such as the mouth and intestines. In severe cases, if Melanoma is not detected and treated early, it can travel through the blood from its origin to other areas of the body.</u>

Diabetic Retinopathy: A diebetic complication which occurs in the eye and might sometimes show no severe symptom, however it damages the blood vessels within the retina and might even cause formation of new blood vessles in the retina. This eye condition could eventually lead to vision loss in case it is not treated early [7], [17], [18].

<u>Glaucoma</u>: Excess pressure at the back of the eye causes damage to the optic nerve and might eventually lead to Glaucoma, a condition which can cause vision impairment and in some cases permanent vision loss. Glaucoma is considered to be one of the leading reasons of blindness in the world [11], [13].

Hypertension (High Blood Pressure): The elevations in the blood pressure typically does not show severe symptoms. However, early indicators of hypertension often appear in the eye such as appearance of dark spots on retina, bleedings in the back of the eye and narrowing the ocular blood vessels [11].

<u>Retinal Detachment</u>: : this condition happens once the retina is pulled or lifted away from its original position. Retinal detachment becomes a severe condition if not treated properly and could eventually casue permanent blindness.

In the next sub-sections two major eye diseases; diabetic retinopathy and glaucoma, are explained in more details. These diseases are detectable through retinal images.

2.2.1. Diabetic Retinopathy

Over time, too much sugar in the blood can lead to blockage of tiny blood vessels that nourish the retina, cutting off its blood supply. As a result, the eye grows new blood vessels; however, these new blood vessels do not develop properly and can easily leak [2]. Diabetic retinopathy is a diabetic complication that affects the eyes and is usually bilateral [10]. Some of the symptoms of diabetic retinopathy include dark strings floating in the vision, blurred vision, fluctuating vision, impaired color vision, dark or empty areas in the vision or even vision loss [19].

In the early stages of diabetic retinopathy, new blood vessels do not grow (proliferate). This stage is called Non-Proliferative Diabetic Retinopathy (NPDR) [20]. In this stage, retinal blood vessel walls would weaken which results in formation of smaller vessels from the vessel walls. These small vessels sometimes leak fluid and blood into the retina. As the number of blocked blood vessels increase, NPDR can progress into a severe state and lead to swelling of nerve fibers in the retina. Sometimes even the macula begins to swell which is a more severe condition and needs immediate treatment [20].

The more severe type of NPDR is known as Proliferative Diabetic Retinopathy (PDR). In PDR condition, the damaged blood vessles are blocked and this causes the growth of abnormal blood vessels in the retina. The newly formed vessels could cause leakage of blood and fluid into the center of the eye [21]. The blood and fluid leakage causes the retina to detach from the back of the eye. Newly formed blood vessles could also interfere with the normal fluid flowing into and out of the eye which causes excess pressure to build up in the eyeball. Excess pressure can damage the optic nerve and eventually lead to glaucoma [10].

2.2.2. Glaucoma

Glaucoma is caused by damage to the optic nerve and could lead to vision impairment and even blindness [13]. Glaucoma is predicted to affect 80 million individuals by 2020; therefore, it is very important to find a method for monitoring the disease and preventing it from permanently damaging the eye [8]. Glaucoma has a critical impact on global blindness. It has been recognised that elevated intraocular pressure exerts direct mechanical damage to the optic nerve head. However, among glaucoma patients, only one-third to half have elevated intraocular pressure at the initial stages. In some, visual field loss continues despite adequate control of the intraocular pressure to normal levels [11].

Both static and dynamic properties of the retinal microcirculation may be implicated in the vascular phenomenon in glaucoma. Study of the retinal microcirculation is thus made possible by the accessibility of retinal vasculature via non-invasive means [11]. Over the past two decades, semiautomated software systems have enabled reliable quantification of geometric components of the retinal vasculature from retinal photography, including retinal vascular calibre, tortuosity, branching angle and fractal dimensions [13], [18].

Advances in technology have tried to resolve the limitations of existing instruments. Peripapillary capillaries is a highly specialised vasculature that supply the nerve fibre layer. The nerve network may indicate the defects in the disc capillary or the damages in the retinal ganglion cells [22]. Studies found consistent associations between quantitatively measured retinal vascular calibre with prevalence of glaucoma and larger vertical CDR [23]. Eye studies showed significantly thinner retinal arteries but insignificant difference in retinal vein diameters. It has been shown that the narrowing of retinal vessels resulting from the glaucoma process is irrespective of the status of angle closure. Reduced retinal nerve fibre layer thickness, greater CDR and characteristic visual

field defects are hallmarks of glaucomatous optic neuropathy. One of the pathognomonic features of glaucoma is the increase of CDR. The increase in CDR is a clinical indicator for glaucoma progression [13].

Imaging systems that employ adaptive optics, such as retinal fundus camera [24], OCT and scanning laser ophthalmoscope [2], have provided in-vivo, high-resolution imaging of the vasculature and nerve fibre layer that overcome poor lateral resolution in conventional ocular optics. These methods have been shown to be in precise agreement with histology in primate studies.

2.3. Retinal Fundus Imaging

Fundus imaging is the photography of retinal semi-transparent tissues using reflected light illuminated from an external source [2]. The image intensities represent the amount of a reflected quantity of light. This is a challenging imaging technique because retina is not illuminated internally and there is a need for an external illumination projected into the eye. The external light and the light reflected by the retina must traverse the pupillary plane. The small size of the pupil in the iris has always made fundus imaging to be technically challenging; therefore, fundus imaging historically involved relatively expensive equipment and highly trained ophthalmic photographers. Over the last ten years or so, there has been major effort around making fundus imaging more accessible [1], [19], [25].

Fundus photography can be performed with colored filters, or with specialized dyes including fluorescein and indocyanine green. Low power microscopic cameras are known as fundus cameras which have ophthalmoscopic design [14].

Fundus photographs enable ophthalmoscopic representation of the patient's retina and are used in detecting some chracteristics of diabetic retinopathy such as microaneurysms and macular edema [21]. This is because retinal details may be easier to visualize in stereoscopic fundus photographs as opposed to direct examination. Fundus photography is also used to help interpret fluorescein angiography because certain retinal landmarks visible in fundus photography are not visible on a fluorescein angiogram [7], [26].

2.4. Optic Disc and Optic Cup Detection

The OD and OC are two key structures of retina. Any change or abnormality in the structure of OD or OC will change the appearance of retina and can be analyzed for diagnosing diseases such as glaucoma [27]. Ophthalmologists use retinal color fundus images to check for any abnormalities in the retina.

There are various image processing approaches for proper segmentation of retinal fundus images to extract OD and OC boundaries. Cup-to-disc ratio is then used for predicting the presence of diseases such as glaucoma [28]. Figure 2.2 [29] shows fundus images and the optic disc and optic cup from a normal eye and an eye with glaucoma.

In the past two decades there were much more algorithms proposed to segment OD than to detect OC. In 1999, an algorithm was proposed by Sinthanayothin et al. in [15] to detect OD by identifying the area with the highest intensity variation. In 2001, an image processing algorithm was proposed to localize the OD in low-resolution color fundus images [5] based on Hausdorff template matching technique. In 2004, a template matching and segmentation model using a deformable contours was proposed [30] to detect OD. In 2010 OD boundaries within retinal fundus



Figure 2.2. The Cup-to-Disc Ratio in (a) normal, (b) Glaucoma retina [29]

images were detected by an extended version of snake algorithm using an implicit active contour [31].

Muramatsu et al. compared three methods of segmentation of OD regions on retinal fundus images: Active Contour Model (ACM), FCM clustering and the Artificial Neural Network (ANN). Three ophthalmologists determined the OD boundaries and provided the diagnosis for the cases. Their results were used as the gold standard for comparison. The average overlap between the results of the proposed algorithms on two different databases and those from the experts were 0.87 and 0.88 for the ACM, 0.89 and 0.88 for the ANN, 0.86 and 0.86 for the FCM. The ANN and ACM methods showed better results and were better options for segmentation of OD for further analysis [26]. In 2012, a curvelet-based algorithm was proposed to detect OD and exudates on low contrast

images [9]. Another OD detection algorithm was proposed to detect OD and extudates on row contrast images [9]. Another OD detection algorithm was proposed in 2012 based on pyramidal decomposition entropy measure, edge detection, and Hough transformation [32]. In 2013, a method was utilized to localize OD based on thresholding. In this method all bright regions are segmented by some area and density criteria [33]. Optic cup segmentation is much more challenging than optic disc detection mainly because there are blood vessels concentrated in OC region, and the intensity level changes from the cup to the rest of OD very gradually. In addition, glaucoma changes the shape of OC and makes is even more difficult to segment it.

Damon et al. proposed a method in 2012 based on vessel kinking to detect OC [34]. The first step was to detect vessels by introducing a segmentation technique. In the proposed method, a SVM classifier was used to detect smaller vessels based on features extracted from the green channel. Then Gabor filter and Canny edge detector were applied to find all possible vessels. Finally, kinking in the vessels was localized and maximum curvature on the vessels were estimated to find the OC boundaries. The overall conclusion of this work was that kinking detection can help improving OC boundary localization [34].

A gradient-based method was proposed by Ingle et al. [35] for OC segmentation in 2013. In this method, the intensity variation in the fundus images were obtained using gradient of the image. Both linear gradient and radial gradient were used. Then CLAHE was applied to RGB components and a threshold was defined to capture OC.

Yin et al. [36] introduced a method based on a statistical model, circular Hough transform, and optimal channel selection to segment OD and OC. In this method, first the OD center was located then its size was approximated. The boundaries of OD was then segmented by Canny edge detector and CHT. On the other hand, OC boundary was extracted by analyzing the green channels and removing the vessels. Assuming that the OC center is close to the OD center, the OC center was extracted by employing an active shape model. The proposed method showed the average Dice coefficient of 0.92 and 0.81 for OD and OC segmentation, respectively [36].

A method was introduced by Ho et al. that used Canny edge detector to find all edges, such as the blood vessels, in the green channel of fundus images [37]. This method then used fast marching

method, inpainting and thresholding to segmenting the image into three regions. In order to estimate the disc boundary, these regions were fitted with two circles. Then an active contour model was applied to extract the boundaries of OC and OD. Using a dynamic histogram equalization, the unclear OC was removed from the results [37].

Rajaiah et al. [27] proposed a method for diagnosis of glaucoma that involved three main stages: OD boundary detection, OC segmentation, and CDR calculation. In order to detect the OD boundaries, Linear Discriminant Analysis (LDA), Contrast Limited Adaptive Histogram Equalization (CLAHE), in-painting and Morphological Operations were used. Segmentation of OC was then performed in green channel of the image by using Watershed transformations. Finally, CDR was calculated to predict glaucoma. The algorithm was tested on 50 retinal fundus images with the success rate of 96% [27].

Thorat et al. proposed an approach for segmentation of OD and OC by using super-pixel classification method [23]. First, Adaptive Histogram Equalization (AHE) was applied on fundus images for preprocessing, image enhancement and removing the uneven illuminations. Then for segmenting the OD, clustering techniques were used to cluster each superpixel as disc or non-disc. For segmenting the OC, thresholding and Gabor filter techniques were applied in addition to the clustering algorithms. CDR was then calculated for further glaucoma screening. The algorithm was tested on 2326 images from 2326 different eyes with promising results [23].

The most problematic issue in OD and OC segmentation is finding the region of interest. Devasia et al. resolved this problem by proposing an automated algorithm for OD boundary extraction by using Fuzzy C-Means (FCM) clustering combined with thresholding techniques [28]. The method was tested on 110 images from DRION database to find the ROI and the results were compared to the work done by ophthalmologists. Although in general the results were promising, there were

still some problems with very bright images where the method could not obtain very accurate results [28]. In another study, Devasia et al. proposed an automated algorithm for OD localization in retinal fundus images [38]. Histogram based Particle Swarm Optimization (PSO) techniques were used for OD detection. The method was tested on images obtained from DRION public database. The total number of test images was 235. The tested images had different histograms with varying OD shapes and sizes. Using the scatter plot and ground truth analysis, the performance of the proposed method was evaluated. The method showed reasonable accuracy and promising results [38].

Sakthivel et al. proposed a method for early detection of glaucoma by using magnitude and phase histogram features from digital retinal fundus images [39]. Feature extraction was done using Local Binary Patterns (LBP) and Daugman's algorithm. After analyzing the histogram features, the Euclidean distance between the feature vectors was used for the prediction of glaucoma. The method was time efficient, reliable and accurate [39].

Chapter 3

Retinal Fundus Image Processing

In this chapter, the steps towards retinal fundus image processing are explained. In section 3.1, the developed method for automatic detection of OD is presented. This section includes subsections covering Contrast Limited Adaptive Histogram Equalization (CLAHE), Alternating Sequential Filters (ASF), and Circular Hough Transform (CHT). In section 3.2, the proposed classification method for detection of OC is described. This classification method indicates those parts of OD which belong to OC. In this section the proposed features extracted from the image are explained. It also includes subsections covering ensemble modeling, Random Forest, GBM, and XGBoost algorithms.

3.1. Optic Disc Detection

In the proposed algorithm for OD detection, five main steps have been performed. A summary of these steps is shown in Figure 3.1. These steps are summarized in the following paragraphs and are explained in more details in the next sub-sections.

The first step is to separate the RGB channels of the fundus image, then to normalize the red, green and blue channels. In this study, all three channels of the color fundus images and their corresponding gray-scale images have been investigated thoroughly using different algorithms to detect OD. The best results were obtained from the red channel where OD region is brighter than rest of the image. In this step, CLAHE algorithm is also applied on the normalized red image for improving the quality of the image and making OD boundaries bold.

In the second step of OD detection, ASF method is applied on the red channel. ASF is an image processing reconstruction method based on sequential morphological operators that is very effective in removing thin unwanted elements within the image [40].

In the third step of OD detection, an adaptive thresholding method is applied to the resulted image from ASF algorithm. In this step, OD region gets separated from other regions of the eye in the fundus image. Forth step is to find the best circle matched to OD region using CHT. The result of this step is a circle that locates OD and creates a mask that segments OD from rest of the image. In the fifth step, some features are extracted from OD in order to further get leveraged in a classification algorithm and to enable capturing and detection of OC.

In the following sub-sections the above-mentioned steps are explained in more details.



Figure 3.1. The flowchart of the proposed method for segmentation of optic disc in retinal color fundus images

3.1.1. RGB Channel Extraction, Normalization and CLAHE

As mentioned in the beginning of Section 3.1, in the proposed algorithm, red, green, and blue channels are first extracted from the color fundus image. Red channel is used for OD segmentation. After normalizing the image, CLAHE is applied on the normalized red image. CLAHE is an image-processing technique used for local contrast improvement and edge enhancement [41]. This method is adaptive meaning that it computes histograms of several distinct regions of an image and uses them for a more even histogram distribution over that image [42]. Overlapping of blood vessels with OD in the region of interest makes the segmentation process very challenging. The goal is to equalize the image histogram such that OD could be easily segmented from the surrounding blood vessels and other parts of the eye in the fundus image.

Contrast Limited Adaptive Histogram Equalization (CLAHE) is an image processing method used in image contrast enhancement. The implementation of CLAHE is different from general Histogram Equalization since CLAHE, as apparent in the terminology, is an adaptive method which means it generates a number of distinct histograms on distinct regions of the image (e.g. small blocks of 8×8 or 16×16 pixels) and uses them all to redistribute image lightness values [41]. This is the main difference between CLAHE, as an adaptive histogram equalization, and the conventional histogram equalization method in which the histogram of the whole image is equalized as one block. Therefore, CLAHE is suitable for improving the local contrast and enhancing the definitions of edges in each region of an image [41].

Since in a small block of image, histogram is limited to small amounts of contrast, noise can be amplified. In order to avoid noise amplification in CLAHE, the contrast limiting method is applied, which means that if any histogram range is above the specified contrast limit, those pixels are

clipped and distributed uniformly to other histogram ranges before applying CLAHE. In this method, bilinear interpolation of blocks is applied to remove any artifact in borders of the blocks.

3.1.2. Morphological Operators: Alternating Sequential Filtering

In many image processing studies, morphological operators have been used with purposes such as filtering, marking and image segmentation [40]. In these operators, the difference between local extrema are calculated in the neighborhood of the given points in an image by defining a disc-shape Structural Element (SE). In this project, ASF is used to remove the unwanted remaining noise resulted from CLAHE algorithm. ASF is a technique which combines opening and closing morphological operations into one algorithm. Image opening operation is once erosin of the structural element occurs followed by its dilation. Image closing operation is defined once dilation of the structural element occurs followed by its erosion. The definitions of opening, closing and ASF are presented in the following equations [40]:

$$A \circ SE = (A \ominus SE) \oplus SE, \tag{3-1}$$

$$A \bullet SE = (A \bigoplus SE) \ominus SE, \tag{3-2}$$

$$ASF^{n}\{A\} = \left(\left(\left(\left(\left(A \bullet SE_{1} \right) \circ SE_{1} \right) \bullet SE_{2} \right) \circ SE_{2} \right) \bullet \dots \right) \bullet SE_{n} \right) \circ SE_{n}, \tag{3-3}$$

where \ominus , \oplus , \circ and \bullet are erosion, dilation, opening and closing operators respectively. In Eq. (3-3), *n* is the order of ASF, and *SE*₁, *SE*₂, ..., *SE*_n are increasingly expanding disc-shape structural elements. ASF results in the removal of gaps in OD mask. Size of SE increases until the best results are obtained. Increasing the size of SE could result in a better OD segmentation up to a certain point. ASF mask is created and applied on the image to extract the OD region. Resulting images from the application of ASF and thresholding methods are then combined together.

3.1.3. Thresholding

The morphological gradients have been used along with thresholding in various image processing tasks [40]. In this step of OD detection algorithm, a threshold is defined to produce a mask separating OD from rest of the image. The value of the threshold is selected for each fundus image authomatically. Since OD is a small portion of the fundus image and it is the brightest region in the red channel, the proposed criteria for threshold selection is to search for a value that keeps less than 10% of the whole image as white (or 1). This value is searched within the intensity range of 0.9 and 0.95 in the normalized red channel after ASF. The smallest value that satisfies the proposed criteria is then chosen.

3.1.4. Circular Hough Transform

Next step is to apply Circular Hogh Transform (CHT) on resulted masked image after thresholding. CHT is a feature extraction algorithm for detecting circles in an input image regardless of their level of perfection. Hough Transform is generally used in shape recognition [43]. Recognizable shapes include lines, corners, ellipses, circules and any other arbitrary defined object. This method is based on a voting rule in order to detect the candidate of best fit for the defined objects and has proved to be robust to noise.

After application of CHT method, centers of all circles are located and their radii are estimated and used to implement a circle of best fit for extracting OD region on the retinal fundus image. CHT method tries to find the circle of best fit in an image based on the general equation of a circle in the x,y plane, as represented in Eq. (3-4).

$$(x-a)^2 + (y-b)^2 = r^2$$
(3-4)

The parameters *a* and *b* are the location of center of the circle, and *r* is the radius.

CHT method is applied on gray scale version of an image in (a, b, r) three-dimensional parameter space. A number of steps are required in order to find the circle of best fit on an image [44]. First, in order to find the most probable circle centers the following steps are required:

- Equation (3-4) is transformed into another equation in two dimensional space (*a*,*b*) to find a fixed value for the radius.
- A specific intensity value is defined, and any image intensity value greater than the predefined number is mapped to (a, b) space in order to find all the possible circles with fixed radius (obtained in the previous step) which can cross the point (x, y).
- High intensity intersections in the (a,b) space are used to estimate the most fit circle center coordinates (a_0, b_0) .

Next, in order to find the best possible radius, the above steps are repeated while radius is not being fixed anymore and will be changed within a specific range.

3.1.5. Gabor Filter

In this project, Gabor filter was used as a feature extraction method in the proposed OC classification method.

The two-dimensional Gabor filters is a popular image processing technique generally used in edge detection, feature extraction and texture analysis. Gabor filters are bandpass filters which allow a certain range (band) of frequencies to pass and block the rest of the frequencies. Gabor filter bank is represented by a mask meaning an array of pixels with assigned values to each of the pixels known as weights. This mask slides over all pixels of the image, and responds to edges and image texture changes through a convolutional process. Equation (3-5) represents the general form of Gabor filter for the purpose of creating Gabor filter bank [45]:

$$g(x, y; \lambda, \theta, \Psi, \sigma, \gamma) = \exp\left(-\frac{x^{\prime 2} + \gamma^2 y^{\prime 2}}{2 \sigma^2}\right) \exp\left(i(2\pi \frac{x^\prime}{\lambda} + \Psi)\right)$$
(3-5)

where $x' = x \cdot \cos\theta + y \cdot \sin\theta$ and $y' = -x \cdot \sin\theta + y \cdot \cos\theta$, λ is the wavelength of the sinusoidal factor, θ is the orientation of the normal stripes to parallel stripes, Ψ is the offset phase, σ is the Gaussian standard deviation and γ is the spatial ratio.

For the purpose of edge detection in this project, two Gabor filter banks were created, each of them including 18 filters with angels from 0° to 170° with the step size of 10°. The parameters used for the filter banks are $\lambda_1 = 7$, $\gamma_1 = 1$, $\Psi_1 = 3$, $\sigma_1 = 2$ for filter bank 1 and $\lambda_2 = 7$, $\gamma_2 = 1$, $\Psi_2 = 3$, $\sigma_2 = 3$ for filter bank 2. Filter bank 1 includes filters with smaller and smoother edges, and filter bank 2 contains filters with longer and sharper edges.

3.1.6. Feature Extraction

In feature extraction step, several features are defined from the detected OD. The goal is to use these features for classifying OD pixels into OC and non-OC. Red, green, and blue channels as well as grayscale version of the images are used in order to define features. Location of the pixels and their distance from the center of OD are also used. The extracted features can be categorized into the following groups:

- 1- Features related to the location of pixels including actual and normalized distances from the center in x and y directions, and Euclidean distance.
- 2- Features related to the color of pixels including actual and normalized intensity values of red, green, and blue channels and gray level in grayscale version of the images.
- 3- Features extracted from CLAHE algorithm: These features are indications of histogram level of pixels in comparison with local neighbors for red, green and blue channels.

- 4- Features extracted from two sets of Gabor filters: These features are indications that a pixel belongs to whether a smooth region or a sharp edge. There are two different sets of Gabor filter banks used in this study with the focus on longer or sharper edges. Both of the Gabor filter banks are applied to the red and green channels.
- 5- Features extracted after applying ASF on red, green and blue channels: In these features, holes and unwanted noises are reduced.

Total amount of 24 features are extracted based on above-mentioned methods. These features are used in OC segmentation in fundus images.

3.2. Classification and Optic Cup Detection

The proposed method for OC detection is based on a classification algorithm. In this study, the extracted features are used for modeling. Five different modeling algorithms have been studied throughout this research including three ensemble methods. The proposed ensemble models are based on Random Forrest, GBM, and XGBoost algorithms. The results from the ensembles are evaluated against the results from Logistic Regression and Support Vector Machines. In the following sub-sections, these methods are explained in more details.

3.2.1. Conventional Methods

As metioned two conventional methods are used in this project for the evaluation purpose of the ensemble methods. In this sub-section Logistic Regression and SVM are briefly explained.

3.2.1.1. Logistic Regression

Logistic Regression is a machine learning technique used in predicting the probability of a certain observation to belong to a certain class using a number of independent predictors (features). The
dependent target variable in logistic regression model is a binary variable. The following is the logistic regression equation, Eq. (3-6), for modeling the conditional probability, where x is the input data, y is the output class label and w is a vector of the weights [46]:

$$P_w(y = \pm 1|\mathbf{x}) = \frac{1}{1 + e^{-yw^T x}}$$
(3-6)

Given a binary class prediction problem $\{x_i, y_i\}_{i=1}^l$, $x_i \in \mathbb{R}^n$, $y_i \in \{1, -1\}$, logistic regression model tries to achieve the minimum value for the following regularized negative log likelihood function, as in Eq. (3-7), where C > 0 is the penalty parameter [46]:

$$P^{LR}(\mathbf{w}) = C \sum_{i=1}^{l} \log(1 + e^{-y_i \mathbf{w}^T x_i}) + \frac{1}{2} \mathbf{w}^T \mathbf{w}$$
(3-7)

Assuming we have two classes, the input data should be linearly separable into the two regions by a linear boundary known as linear discriminant. Type of the linear boundary (straight line, plane, etc.) is decided based on the input data points. Following is the equation of the linear discriminant function, where x_1 and x_2 are the input variables:

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 \tag{3-8}$$

Given a data point location as (a, b), there are three possible outcomes:

- 1- The point lies within the region which is predefined by the positive class observations. In this case, the plane would be positive and lie somewhere within $(0, \infty)$ region. Higher magnitude of this value indicates higher probability of the point (a, b) belonging to the positive class.
- 2- The points could lie in the regions defined by negative sample points. In this case, the boundary plane would be negative and lie somewhere within $(-\infty, 0)$. Higher magnitude

(absolute value) of this value indicates higher probability that point (a,b) belongs to the negative class.

3- The point lies exactly on the boundary plane. In this case, the equation of the plane will be equal to zero and the outcome probability value will be 0.5 indicating that there is an equal chance that point (*a*, *b*) belongs to either of the two classes.

One important point to consider in logistic regression modeling is overfitting, while adding more features to predict the target variable. Adding too many features would make the model learn the behavior of the data on which it is being fit, and therefore the generalizability of the model will be reduced [46].

3.2.1.2. Support Vector Machine

Support Vector Machine is a machine learning algorithm which is used mainly for classification problems. In this technique, each data record is considered as a point in *n*-dimensional space, *n* being the number of features, and the algorithm tries to classify the data points by obtaining the best hyper-plane which separates the data points well into defined number of classes [47]. The main goal of SVM is to find the best hyper plane to classify the data points into the defined number of classes. The algorithm has a built-in capability to ignore outliers and find the best hyper-plane with maximum margin. Therefore, SVM is proved robust to outliers. One of the issues that might come up is once the data is not separable. SVM has a method known as the kernel trick. Kernel function takes a low dimensional input space and changes it to a higher dimension space to make the data separable. There are multiple kernel functions available such as linear, rbf and poly kernels where rbf and poly are kernels for constructing non-linear hyper planes.

There are some pros and cons in applying SVM model. Associated benefits include but are not limited to: high performance in case of clear separation margin, high performance once the number of features is greater than the number of observations, memory efficiency. The associated costs include but are not limited to: poor performance in case of a large dataset due to high required training time, poor performance in case of noise and overlap in the observations [47].

3.2.2. Ensemble Methods

The main goal of ensemble machine learning algorithms is to improve the prediction performance by combining the results from individual algorithms [48]. In machine learning, ensemble methods are among supervised learning methods that use multiple learning algorithms instead of one individual principal learning algorithm. Two main benefits of using ensemble models are obtaining improved prediction performance and producing more stable models [49]. In general, a machine learning ensemble consists of a finite set of alternative models with flexible structures to search through the data space in order to find the most suitable predictions for a particular modeling problem. The ensembles combine multiple algorithms to form a better result which might be missed by a single learner. Ensemble methods can be considered as methods to compensate poor performance of a single learner by executing a lot more computation. It should be emphasized that it is sometimes very difficult to obtain a good performing solution for some specific modelling problems and there is not always a well-suited solution for every particular modeling case.

Training and evaluation of an ensemble is typically more complicated compared to a single model and an ensemble needs more computation effort. Therefore, simpler and faster algorithms are usually used in an ensemble rather than slower ones to improve training and validation speed.

27

Common types of tree-based ensembles are Bootstrap Aggregating (bagging) and Boosting [50], [51].

In the bagging method, each model in the ensemble is trained by a randomly drawn subset of the training set, and all models vote with an equal weight [50]. Bagging involves creating multiple copies of the original training data using the bootstrap, fitting a separate decision tree to each copy, and then combining all trees in order to create a single predictive model. Random Forest algorithm is an example of bagging family which combines random decision trees. In Random Forest, each tree is built on a bootstrap dataset and is independent from the other trees. This method usually has high classification accuracy [49], [50], [52].

In boosting algorithms, the ensemble is incrementally built by emphasizing on the misclassified training instances. A popular example of boosting ensemble is Gradient Boosting Machines [53]. In GBM, the trees are grown sequentially using information from previously grown trees. Boosting does not involve bootstrap sampling; instead each tree is fit on a modified version of the original data set to reduce the misclassification.

3.2.3. Random Forest

Random Forest is a popular ensemble modeling algorithm based on multiple decision trees. A Random Forest model combines decision trees that are trained on subsamples (bootstraps) of the training data [50]. The decision trees in the forest are, therefore, slightly different from each other. Then results of the decision trees, as the learners, are combined by a voting system with the same weighting for all decision trees. Since a Random Forest takes many uncorrelated learners to make the final model, it reduces both error and variance. In most cases Random Forest method has proved to have a better performance compared to individual decision trees and individual models [49].

3.2.3.1. Decision Tree

The goal of a decision tree is to create a predictive model to estimate the value of a target variable by learning simple decision rules captured from the feature space. Decision trees partition the feature space into a set of rectangles and consider simple decisions for each of these partitions [54]. Decision tree based models are able to handle both numerical (regression trees) and categorical (classification trees) data.



Figure 3.2. Decision Tree

When a decision tree is trained, the data is subjected to a series of criteria that evaluate the features to determine outcomes (labels). The idea is to split the data into smaller pieces in a sequential manner based on the descriptive features until all the data samples fall under one outcome. In case of classification trees, the outcome takes values such as 1, 2, ..., K. In decision trees, the complete dataset is located at the top of the tree. The trees consist of a root node and a number of leaf nodes. Any observation that satisfies the condition at a single node, is assigned to the left branch, and the rest of the data is assigned to the right branch.

The trees are usually subjected to a limited depth in order to reduce the chance of overfitting. The maximum depth is usually predefined in the initial modeling stages.

The partitioning process results into grouping similar observations in the same group. This process is done in a recursive manner and the data is split into two groups at each iteration. The process continues until some stopping critera are reached. Some examples of these criteria could be once all the data in the same group express a defined similarity, or once the depth of the tree reaches a maximum level without getting overfitted to the training data.

The training procedure of a decision tree is to select the best conditions in order to split the data into the most similar (homogenous) groups. This is done by introducing enthropy as given in the below equation where p_i is the ratio of elements in label *i* to the size of all data:

$$Entropy = \sum_{i=1}^{c} -p_i \log_2(p_i)$$
(3-9)

Entropy is a value between 0 and 1; it is a measure of impurity in a dataset after grouping. A perfect decision is when the entropy is equal to 0. When all the samples in the same group belong to the same class, the entrophy is 0. Therefore, the training procedure tries to minimize the entropy of the new introduced groups at each iteration.

3.2.3.2. Bootstrap aggregating

Random Forest is a learning algorithm which works in a parallel mechanism meaning that each decision tree is generated independently of other trees. Each individual decision tree is constructed using a subspace of either features or input data. Therefore, each tree normally has different sets of characteristics based on if it is constructed using random features, random input data or a combination of these.

Bootstrap is a process which generates several new training data sets from the original training data. Considering $D = \{(x_n, y_n), n = 1, ..., N\}$ as the original training data with the size of N samples, the goal is to generate M number of bootstrap training sets, called D_j (j=1,...,M) in order to train M number of models in the ensemble where $D_j = \{(x_n, y_n), n = 1, ..., N_j\}$. Bootstrapping process is sampling with replacement which is a sampling method where each item is replaced immediately after being selected and each sample might have duplications of the same observation. Therefore, each new dataset generated by bootstrapping method might contain duplicated observations. Normaly, the input observation x_n are vectors whereas the outputs are class labels. The features selected in each bootstrap D_j might include all or in some cases partial features from the original training set D.

Aggregation is the voting process over all trained models in order to calculate the final decision of the ensemble for an input vector \mathbf{x} . In Random Forest algorithm, the aggregation method is usually based on averaging or voting by all trees with equal weighting system. Regardless of how the decision tree is constructed, the following formulation is applied in order to obtain the probability of an input \mathbf{x} belonging to class c:

$$P\left(c|v_j(\boldsymbol{x})\right) = \frac{P\left(c|v_j(\boldsymbol{x})\right)}{\sum_{k=1}^{n_c} P(c_k, v_j(\boldsymbol{x}))}$$
(3-10)

where $v_j(\mathbf{x})$ is vote by tree j, and n_c is the number of classes [55]. In the Eq. (3-10), the output is estimated by the function of class c points over all points that are assigned to in the training set. The discriminant function is then defined as in Eq. (3-11):

$$g_{c}(\mathbf{x}) = \frac{1}{N} \sum_{j=1}^{N} \hat{P}(c|v_{j}(\mathbf{x}))$$
(3-11)

where $g_c(x)$ is an indication of the votes for input *x* to be in class *c*. The decision obtained from random forest for an input *x* is the class in which the discriminant fuction is maximum for *x*. If all trees are fully split, this decision rule is equivalent to the plurality vote among all trees.

3.2.4. Gradient Boosting Machine

GBM is a machine learning technique for regression and classification problems. In this method, a prediction model is produced based on an ensemble of weak predictors. The principle idea of boosting is to add new predictors to the model sequentially by emphasizing on the misclassifications in each sequence [53].

The original GBM algorithm was proposed by Friedman [56]. In this method, the first predictor is trained on the whole training data set with equal weights to each observation. The next predictors are trained on the training set based on performance of previous predictors. At each particular sequence, the misclassified observations get higher weights by the algorithm and a new weak learner is added to the model with respect to the error of the whole trained ensemble. The training process continues to add classifier learners until reaching a specific accuracy or until the loss function reaches its minima. In GBM algorithm, new learners are constructed to be maximally correlated with the negative gradient of loss function of the whole ensemble. One of the major issues of GBM is overfitting to the training data [57].

The GBM algorithm can be summarized as in the following:

- First, a simple predictor (decision tree) is trained on the whole data and the residual error is calculated. Data points with the largest residual error are normally difficult to be fit in a simple ensemble.
- Then, a new predictor is added to the ensemble on the misclassified data points. The new predictor uses the same input features and residual errors are targeted. The main goal is to target the misclassified data points and to reduce the residual errors.
- New predictors are fit to the remaining residuals and added to the ensemble until the sum of the residuals become constant or the loss function reaches its minima. New predictors are added to the ensemble and are focused on misclassified data points to get them right.

3.2.4.1. Loss Function

Considering $D = \{(x_n, y_n), n = 1, ..., N\}$ as the training data with the size of N samples, the goal of training procedure is to add more trees to GBM ensemble to minimize a loss function. Loss function is defined to monitor the training progress and the predictions. In the gradient boosting procedure, the goal is to find a classifier to map the input vectors x to the target output y. Loss function, L(y, F(x)), could be defined with a variety of functions. In GBM, it is critical to

$$L(y,F(\boldsymbol{x})) = \frac{1}{2} \sum (y - F(\boldsymbol{x}))^2$$
(3-12)

$$L(y,F(\boldsymbol{x})) = \sum |y-F(\boldsymbol{x})|$$
(3-13)

For the training data *D*, the boosting algorithm expands the model to minimize L(y, F(x)). $F^*(x)$ is considered as the best expected function that minimizes the loss function as:

define a differentiable loss function. The most common functions are the following:

$$F^*(\boldsymbol{x}) = \arg\min_{F(\boldsymbol{x})} E_{\boldsymbol{x},\boldsymbol{y}} L(\boldsymbol{y}, F(\boldsymbol{x}))$$
(3-14)

where $E_{x,y}$ is the expected value of loss function. In GBM training procedure, in each iteration a single tree is added to the model to decrease $F^*(x)$ until it reaches to a minimum value.

3.2.4.2. GBM Training Procedure

Since GBM is trained in a sequential manner, the model at each iteration depends on the previous iteration. The trained model at iteration M is shown by $F_M(x)$. In general F_M is the resultant prediction by all trees in the ensemble which could be written as:

$$F_M(\boldsymbol{x}) = \sum_{m=0}^M \beta_m h_m(\boldsymbol{x}; a_m)$$
(3-15)

where h_m is the decision tree and β_m and a_m are the expansion coefficient and the parameters of m^{th} decision tree, respectively. The gradient of loss function has an important role in the learning process; therefore, the defined loss function should be differentiable:

$$g_M = -\frac{\partial L(y, F_{M-1}(x))}{\partial F_{M-1}(x)}$$
(3-16)

where g_M is the gradient of the loss function used in iteration M and is called pseudo-residuals. The new decision tree can be added to the ensemble using the following equation:

$$(\beta_{M+1}, a_{M+1}) = \arg\min E_{x,y} L(y, F_{M-1}(x) + \beta_m h_m(x; a_m))$$

 β, a (3-17)

In general, the training process of GBM ensemble can be summarized as in the following:

- 1- Initialization:
 - A) Set learning rate
 - B) Fit a tree to dataset D
- 2- Adding new trees to the residuals:

- C) Calculate $L(y, F_0(x))$ to find the residuals
- D) Create a new tree to optimize the loss function
- E) Check if the stop criteria is achieved.

3.2.5. Comparison between Random Forest and Gradient Boosting Machine

Fundamental differences between Random Forest and GBM are presented in this section. In the following, the main advantages of Random Forest over GBM are listed [49], [50], [53], [56]:

- Random Forest can be easily run in parallel distributed systems. In general, GBM runs step by step and not all of proposed GBM versions can be run in parallel systems. Random Forrests are preferred when dealing with large datasets or a large number of trials.
- Subsampling and bagging steps in Random Forest typically improve the performance.
- Random Forest is less sensitive to noise compared to GBM when data is noisy. Since GBM algorithms are built on the residuals, they usually have higher variance with noisy data compared to Random Forest.
- Since Random Forest creates a large number of uncorrelated decision trees, they are resilient to overfitting. The decision trees might get overfitted to the data in a different way, but through the voting system in Random Forest, the overfitted results are averaged out.

In the following, the main advantages of GBM over Random Forest are listed [49], [50], [53], [56]:

- GBM is preferred over Random Forest when size of data is not a concern. This is because GBM algorithms improve the performance by focusing on misclassifications.
- In the training process, GBM adds new decision trees to compliment the previously built ensemble. This normally introduces higher accuracy with fewer trees.

35

- The process of adding a new classifier in GBM in each training sequence could result into an improved ensemble. On the other hand, the classifiers in Random Forest are trained independently. Therefore, GBM usually has preferred training process compared to Random Forest after model tuning.

Figure shows the main difference between Random Forest and GBM in terms of algorithm.



(a)



Figure 3.3. (a) Random Forest vs. (b) Gradient Boosting Machine

3.2.6. Extreme Gradient Boosting (XGBoost)

The Extreme Gradient Boosting algorithm [58], or XGBoost, is a scalable end-to-end ensemble classification method based on the original GBM that was proposed by Friedman. The XGBoost is proposed by Tianqi Chen and Carlos Guestrin as a novel model to build a scalable boosting tree system for sparse data [59]. XGBoost has two major enhancements over other GBM methods: (a) speeding up the tree construction and (b) proposing a new distributed algorithm for tree searching. GBM algorithm is optimized in XGBoost and the learning algorithm can handle sparse data. A theoretically justified weighted quantile sketch procedure made XGBoost capable of handling instance weights in approximate tree learning. Since XGBoost is also capable of parallel and distributed computing, it is much faster then other gradient boosting methods. At the same time, it has the accuracy and efficiency of GBM. It should be noted that XGBoost can process millions of samples on a single computer.

The following steps were proposed by [59] to improve the speed of GBM in XGBoost:

- Weighted Quantile Sketch: One important step in XGBoost algorithm is to propose candidate split points on the data. Usually percentiles of a feature are used to make candidates distribute evenly on the data. In XGBoost, a novel distributed weighted quantile sketch algorithm is utilized that can handle weighted data in large datasets. The general idea is to propose a data structure with merge and prune operations to maintain a certain accuracy level [59].
- Sparsity-aware Split Finding: Data sparsity could be a result of missing data points, frequent zero values, or feature artifacts. XGBoost can handle all sparsity patterns in a unified way by making computation complexity linear to number of non-missing entries in the input.

3.2.6.1. XGBoost Parameter Tuning

XGBoost algorithm has become a very popular and powerful algorithm in dealing with complex data problems. It is implemented using a large number of parameters which makes the model difficult to tune and optimize, at the same time make it one of the best model candidates for addressing complex data problems along with irregularities within the data.

As explained in previous section, XGBoost is an extreme and advanced implementation of GBM algorithm with some built in enhancements. The following section lists some of the advantages of this algorithm along with a short summary for each item:

Parallel processing: although boosting is a sequential process in nature, and each tree is only constructed after the previous tree, XGBoost has an advanced implementation which enables parallel processing, therefore it supports higher training speed.

Missing values: XGBoost has a built-in module which enables handling of the missing values. Its implementation enables taking a missing value parameter from the user and learn through time which path to take in order to handle future missing values.

Flexibility: XGBoost implementation enables custom evaluation and optimization criteria which makes the model very flexible in terms of what a user can achieve.

Regularization: the model has a built-in regularization method which helps in reducing chance of overfitting to the training set. For this same reason XGBoost is also known as regularized boosting algorithm.

Tree Pruning: XGBoost splits the data up to the set maximum tree depth parameter and then it starts pruning the tree and wherever there has been no positive gain, the algorithm removes that specific split. However, GMB algorithm stops the splits whenever it encounters a negative loss function. There might be cases where a positive gain happens after the negative loss. In these cases,

XGBoost will have a high advantage of not missing out on the positive gain just because it has stopped due to a local negative loss.

XGBoost algorithm is implemented using multiple parameters. These parameters fit into 3 categories: general parameters, booster parameters and learning task parameters.

XGBoost general parameters guide the entire model functionality. Below is a list of some of the general parameters and a brief description for each:

booster: a parameter for selecting the type of the model for each iteration, the default type is *gbtree* which indicates the tree-based model

silent: a mode parameter which affects the printing of messages during run-time. Setting this parameter to zero helps in better understanding the model.

The second parameter category is the booster category which includes parameters that guide the individual tree (booster) construction at each iteration. Below is a list of some of these parameters and a brief description for each:

max_depth: this parameter indicates the maximum depth of each tree. If the depth is set too high, the algorithm learns to adapt too much to that specific sample and therefore would fail to correctly make predictions once used on other samples, a condition known as overfitting. The values are typically set somewhere between 3 and 10.

max_leaf_nodes: this is a parameter which indicates the number of leaves or child nodes in each tree. The parameter typically does not need much tuning since the algorithm can adapt based on the set max_depth value, as an example tree depth of n will construct 2^n leaf nodes.

learning_rate: this parameter helps in creating a more robus model through weight shrinkage on each iteration, this parameter is typically tuned between 0.01 and 0.2.

39

gamma: each node splits once the previous split results in a positive value for loss reduction. Gamma parameter specifies the minimum value of loss reduction required for making the next split. This can create a conservative algorithm and therefore the gamma parameter is better to be tuned.

subsample: this parameter indicates what fraction of data to be sampled for each tree. Setting a low value would prevent overfitting, however if the set value is too low then it might result in under-fitting. Typically, the range of allowed values is set between 0.5 to 1.

lambda: this parameter is set for handling regularization of XGBoost (L2 regularization) in order to avoid overfitting. L2 regularization or Ridge regression is based on the least squared error. It is more sensitive to the outliers since it squares the errors and magnifies them.

alpha: this parameter also deals with regularization (L1 regularization) and in cases where data is very high-dimensional, tuning this parameter would help to build a faster implementation of the algorithm. L1 regularization or LASSO regression is based on least absolute deviation. L1 regularization tries to minimize the sum of absolute differences between the target value and the estimated value. It is robust to outliers.

The third parameter category is the learning task category which helps in optimization of the algorithm. Below is a list of some of these parameters and a brief description for each:

objective: a parameter to define the loss function to be used for tree construction. Some examples could be *logistic* function for binary classification where the returned prediction is a probability or the *softmax* function for multi-class classification problems where the returned prediction is a class and not probabilities.

eval_metric: this parameter sets the metric for evaluating the model performance. Some of the typical examples are *rmse* (root mean squared error), and *auc* (area under the curve)

40

seed: this parameter is sometimes tuned in order to enable reproducing of the same results in multiple run times if necessary

Tuning and optimization of a certain model is very practical in nature, and the more one has experience in modeling, the faster this process will become. In some problems, not all the parameters are necessary to be tuned up and default values result in high performance models. Although the process varies based on which problem is to be tackled; the following is a list of some general rules to follow in the tuning and optimization process:

- The model needs to start with a relatively high value for the learning rate
- High set value of the learning rate is used to obtain the optimum value for the tree depth
- Next step is to tune up the tree specific parameters (booster parameters) such as maximum depth, subsample and gamma.
- Then regularization parameters need to be tuned up if one decides to use the regularization capability of XGBoost. Tuning these parameters (lambda and alpha) helps to enhance model performance and reduce model complexity
- The final step is to change and lower down the value of the learning rate to achieve the optimal parameters and a high performing model

3.2.6.2. Cross Validation

Cross validation is an approach to estimate the performance of a modeling algorithm. The goal is to find the best model among multiple methods. In cross validation the data is split into training and test sets. The training set is then split into k folds, for example 10 folds. The algorithms are validated on the data from each fold. After running cross validation, k different performance scores are obtained and the performance can be summarized based on the mean and standard deviation. The number of folds, k, must be selected such that the size of each test fold is large enough as a

sample size. The results from cross validation method are more reliable and more accurate estimation of the performance of the algorithms is obtained [60]. It is more accurate because the algorithm is trained and evaluated multiple times on different datasets.

3.2.6.3. Grid-Searching

Grid-searching is a model hyperparameter optimization technique. It is the process of scanning for the optimal parameters to build a model on a specified dataset. Certain parameters should be welltuned in the model depending on its type and the application [61]. Grid-Search tests all possible combinations of the parameters under study and builds a model of each combination in each iteration. In this process a model and its evaluations can be stored for each combination. Since grid-searching can be computationally very expensive [61], it is important to search the significant parameters in the model. Grid-searching can be applied across different algorithms to find the parameters that result in the best model.

3.2.7. Classification Method Selection

As mentioned above, five classification algorithms have been studied in this project: Logistic Regression, Support Vector Machines, Random Forrest, GBM, and XGBoost. These methods are trained and validated to classify cup and non-cup pixels within OD.

Figure 3.2 summarizes OC classification process performed in this research.

The following steps are performed in this study to evaluate these methods and compare the results:

- 1- The images are randomly divided into training and testing datasets. 70% of the images are selected as training set and the remaining 30% are used as testing dataset.
- 2- The OD pixels are labeld as OC and non-OC pixels based on the OC boundaries marked by six ophthalmologists as experts.

- 3- All pixels in the training set are randomly mixed.
- 4- Each model performance is evaluated 10 times by randomly selecting 10 folds from the training dataset. Each fold is divided into training (50%) and evaluation (50%) sets. The folds might have overlaps, but the training and evaluation sets in each fold do not have overlap. Every model is trained 10 times and evaluate 10 times using 10 folds.
- 5- The average accuracy and training speed is calculated based on the training/evaluation folds.

After the models are evaluated, each method is trained by the whole training set then tested based on the labels from the experts.



Figure 3.4. OC Classification Algorithms

3.3. Distributed Systems

The algorithms used in this project are implemented in a Spark distributed system to benefit from its fast computational power. A distributed system is a group of computers which work together and appears to the end user as a single computer [62]. Each of these computers can fail without affecting the system as a whole. Management, maintenance and deployment of a distributed system is complex. Distributed systems enable the user to scale the system horizontally. Normally scaling is done vertically meaning that in order to handle traffic we need to upgrade the hardware. Vertical Scaling can boost the performance of the system up to the latest hardware capabilities and therefore it works well only until a certain point. Scaling horizontally means adding more computers instead of upgrading the hardware. In the horizontal scaling case, there is no limit on how much we can scale up the performance. It is also cheaper than vertical scaling after passing a certain threshold. In addition to horizontal scaling, low latency and fault tolerance are the other important benefits of distributed systems.

3.3.1. Different Types of Distributed Systems

A distributed system only works when the software is specifically designed to run on multiple computers at the same time while being able to deal with the problems specific to distributed computing. One example of scaling would be to scale up a database. If a database starts getting increased number of query requests and the number starts getting larger that the number of queries it can handle; the application, a program built in Spark with a driver program and cluster executors, will noticeably decline in performance [63]. One method of increasing read performance from a database is called master-slave replication strategy. In Master-Slave Replication strategy there is additional database servers which are synced with the main database as shown in Figure 3.5. The user can read from the additional databases and insert into the main or master database. This way the application has been scaled up to execute three times more read queries.



Figure 3.5. Master-Slave Replication Strategy

There are a number of distributed system categories currently available each with a number of public production usage. Following is a list of some of the distributed system categories and some details on each:

1- Distributed Data Stores: they are also known as distributed databases and are used extensively. Most of the distributed databases are non-relational or NoSQL databases. These databases are very high performance and scalable but less consistent. There is a famous theorem on why distributed databases cannot be consistent, partition tolerant and available all at the same time. The theory is called the CAP theory which stands for Consistency, Availability and Partition Tolerant, as shown in Figure 3.6. A system is consistent when the sequential input and output is always same as expected. A system is available when the system does not die and unless the node has failed, it always returns a

response. A system is partition tolerant when it continues to be consistent and available in case of network partitions. One publicly known distributed NoSQL database is Cassandra.



Figure 3.6. CAP Theorem: Consistency, Availability, and Partition Tolerance Overlaps

2- Distributed Computing: it is splitting a big task into smaller ones since no single system can execute the original task alone. The smaller tasks is then executed on a number of machines in parallel. Distributed computing enables horizontal scaling by adding more number of nodes to handle bigger tasks. One publicly known distributed computing system is MapReduce that includes data mapping and data reducing to produce an output. There are some issues with MapReduce [64]. One of the issues is batch processing. Since the MapReduce jobs are in batch mode, the whole task needs to be restarted in case one of the jobs fails; this makes the processing task very slow. The other issue with MapReduce is the time it takes to produce results which is a big issue when it comes to real time big data analysis. A number of new tools have been introduced to deal with these issues with combining batch processing and stream processing. A few examples are Apache Spark and Kafka Streams.

3- Distributed File Systems: they allow accessing different files using the same interface. Distributed file systems go hand in hand with distributed computing and distributed databases. One publicly known example is Hadoop Distributed File System (HDFS) which is used for computational jobs ins Hadoop framework [64]. HDFS is used to replicate and store big data files using many machines. As shown in Figure 3.7, HDFS architecture contains data nodes and name nodes. Data nodes are responsible for file storage and command execution such as writing new data. Name nodes are responsible for keeping the cluster metadata such as which file is stored on which node.

Distributed Messaging systyems: they provide central storage space for system events or messages. Examples of distributed messaging systems are Kafka, RabbitMQ and Amazon SQS.



Figure 3.7. Hadoop Distributed File System Architecture [64]

3.3.2. Apache Spark Overview

Apache Spark is a general cluster computing framework which has API implementations in a number of programming languages such as Python, Scala, R and Java. Apache Spark also supports a number of high level tools such as MLlib, Spark SQL and Spark Streaming. Previous versions of Spark supported Resilient Distributed datasets (RDDs); however, after Spark version 2.0, RDDs are replaced by dataframes with a richer optimization implementation in the backend; therefore, it has a better performance [63].

In Spark cluster mode, the Spark applications run on a cluster as completely independent sets of processes. Theses sets are managed by the SparkContext object in the driver program. SparkContext first connects to a number of cluster managers, defined as services used for getting resources on the cluster, such as YARN, Mesos or standalone manager; then, resources are allocated across the processes. When the connection happens successfully, node executors are acquired. Node executors are processes that store data and run application computations. The application code are then sent to the executors and SparkContext sends tasks to each of the node executors [63].

Spark supports a caching mechanism to enable pulling data into in-memory cache. This is a very useful functionality specifically when the data needs to be accessed frequently in an application [63].

Each spark application has separate executor processes for itself. These processes are up while the application is running. The processes run different tasks in multiple threads to benefit from separating the applications and making them independent. It also makes the data not to be sharable on different applications without having the data written to an external storage. The main program, known as the driver program, should always accept connections from its own executors. This

48

means that the main program should be accessible from the worker nodes, as shown in Figure 3.8. Worker node is any node which is able to run an application in a cluster.



Figure 3.8. Apache Spark Cluster Overview [63]

Apache Spark currently supports four types of cluster managers as following:

- 1- Apache Mesos: which is a general cluster manager that can also run Hadoop MapReduce applications.
- 2- Hadoop YARN: it is the cluster manager in Hadoop 2.
- 3- Kubernetes: it is an open source system that automates the scaling and deployment and management of applications.
- 4- Standalone: it is a cluster manager contained within Spark for setting up a cluster.

Spark controls the resource allocation both within and across applications. Within application resource allocation means control over the resources if multiple computations are happening in parallel on a single SparkContext. Across application resource allocation means controlling the resources at the cluster manager level. Different applications can be submitted to any of the above-mentioned type of cluster managers by using spark-submit script [63].

Chapter 4

Results and Discussions

In this chapter, results of the proposed OD and OC detection algorithms are presented. First, results from the proposed OD detection method are shown. Then results of the classification methods for distinguishing OC from non-OC pixels within the OD are presented and discussed.

The images used to evaluate the proposed algorithms are from MESSIDOR database [19] MESSIDOR is a publicly available database that is designed to evaluate segmentation and indexing techniques in the field of retinal ophthalmology. MESSIDOR is part of a research program sponsored by the French Ministries of Research and Defense. It contains images of 2240×1488 pixels in uncompressed TIFF format. In this thesis study, one subset of 100 images was used. This subset includes prime images as well as images with OD and OC boundaries marked by expert ophthalmologists.

4.1. Optic Disc Detection Results

As explained in Chapter 3, the proposed method of OD detection includes several steps. In this section, the results from each step is presented for some sample fundus images. This selection shows strong capabilities of the proposed method in detecting OD in images with normal contrast, images with artifacts, and images with low contrast in the optic disc boundaries. Figure 4.1. shows

the selected fundus images from MESSIDOR database. Figures 4.2, 4.3, and 4.4 illustrate red, green and blue channels extracted from the fundus images.



(a)





Figure 4.1. Six sample retinal fundus images from MESSIDOR database.

A comparison between red, green and blue channels indicates that red and green channels provide better contrast difference than blue channel at the OD boundaries.



(a)

(b)



(d)





As shown in Figure 4.2, red channel has an advantage in OD detection; however, it suffers the most from artifacts in the rest of the image. Figure 4.3 shows that green channel is potentially the best option in detection of vessels, which is not the purpose of this study.



(a)



(c)



Figure 4.3. Green Channel

On the other hand, blue channel may help in better classification of OC. In this research, OD boundaries are detected using the red channel; however, all channels were used in OC classification.



(a)

(b)



(d)



Figure 4.4. Blue Channel

The results from normalization and histogram equalization of red channel images by CLAHE algorithm are shown in Figure 4.5. This figure demonstrates that local histogram equalization helps in fading blood vessels specially in regions close to OD boundaries.



(a)

(b)



(c)



Figure 4.5. Red Channel after CLAHE

The results of using ASF and thresholding are illustrated in Figure 4.6. After thresholding, the black-and-white masks are obtained which are used for the next step in OD detection. In case of artifact appearance in the image, they might still be present in the resulted mask.



Figure 4.6. Red Channel after CLAHE, ASF, and Thresholding

Figure 4.6 (d) is an example of the cases which contain many bright regions in the original fundus image. Figure 4.7 shows the detected OD regions after CHT algorithm. CHT is capable of removing the artifacts from bright regions as shown in Figure 4.7 (d)



(a)





Figure 4.7. Red Channel after CLAHE, ASF, Thresholding, and CHT

Figures (4.8) and (4.9) illustrate comparisons between detected OD boundaries using the proposed algorithm and those manually detected by experts. The figures represent a close match between the results.



(a)



(c)

(f) (e)





(c) (d)





Table 4.1. The fundus images used in this study are from MESSIDOR database and each image has 2240×1488 pixels.

		Pixels Marked by Ophthalmologists		
		OD	Non-OD	Total
Pixels Marked by Proposed Algorithm	OD	True Positive (TP): 1,941,219	False Positive (FP): 159,828	2,101,047
	Non-OD	False Negative (FN): 103,468	True Negative (TN): 331,107,485	331,210,953
	Total	2,044,687	331,267,313	333,312,000

Table 4.1. Evaluation of optic disc detection algorithm

The precision of the proposed OD detection algorithm is 92.4% since the total number of 1,941,219 pixels are correctly marked as OD and the total number of 159,828 pixels are not considered as OD by the experts. The recall, accuracy and f-score of the proposed method are 94.9%, 99.9% and 93.6%, respectively. The equations of these parameters are listed in the following:

$$precision = \frac{TP}{TP + FP} \times 100\%$$
(4-1)

$$recall = \frac{TP}{TP + FN} \times 100\%$$
(4-2)

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%$$
(4-3)

$$f - score = \frac{2 \times precision \times recall}{precision + recall} \times 100\%$$
(4-4)

Precision is the probability that a randomly selected predicted OD pixel is actually in OD. Recall or sensitivity is the probability that a randomly selected actual OD pixel is also predicted as OD. F-score is the harmonic mean of precision and recall. In Table 4.2, the result of the proposed OD detection method is compared with some other studies on MESSIDOR database.
Study	Precision	Recall	Accuracy
Kumar and Sinha [65]	89.5%	93%	
Aquino et al. [66]	86%		99%
Lu [67]	91%-93% *		98.7%
Almazroa [68]	86.6%		
Proposed Method	92.4%	94.9%	99.9%

Table 4.2. Comparison of the proposed method and other studies

* Applied on a different database

As presented in Table 4.2, the results from the proposed method is more accurate than the other researched in detection of OD.

4.2. Optic Cup detection Results

In this section, the OC classification results from five different algorithms are presented and compared. As mentioned in Chpater 3, these algorithms are based on Logistic Regression, SVM, Random Forest, GBM, and XGBoost. To better compare the performance of these methods, the following steps are performed:

- 1- The images from MESSIDOR database are randomly divided into train and test datasets. As mentioned before, a set of 100 images is used in this thesis project. 70 images are randomly selected as the training set and the remaining 30 images are used as the test set.
- 2- The OD pixels detected by the proposed algorithm are used in the modelling. The OD pixels in both train and test sets are labeld as OC and non-OC pixels based on the OC boundaries marked by the experts.
- 3- All the pixels in the training set are mixed for both training and evalution of the models.

- 4- Each method is evaluated 10 times by randomly selecting 10 folds from the training set.
 Each fold contains train and evaluation sets that are used to train the model and evaluate its performance.
- 5- At the end of the evaluation step, the average performance and speed of training for each method is calculated.
- 6- Each method is then trained by the whole training set and tested on the testing images.
- 7- Final results from the test set are compared with the labels from experts.

Tables 4.3, 4.4, 4.5 and 4.6 present the accuracy and training duration for Logistic Regression,

SVM, GBM, and Random Forest models, respectively:

Logistic Regression Model	Training Time	Accuracy (%)	Details	
Validation Fold 1	55 s	88.74		
Validation Fold 2	58 s	88.77		
Validation Fold 3	48 s	88.69		
Validation Fold 4	48 s	88.74		
Validation Fold 5	45 s	88.76	Each validation fold is divided into a	
Validation Fold 6	28 s	88.74	training set (50%) and a test set (50%).	
Validation Fold 7	33 s	88.72		
Validation Fold 8	41 s	88.72		
Validation Fold 9	57 s	88.77		
Validation Fold 10	59 s	88.80		
Average (Validation Folds)	47.2 s	88.75		
All Data	115 s	88.51	Training set: Randomly selected images (70%) Test set: Remaining images (30%)	

Table 4.3. Evaluation of Logistic Regression Algorithm

Support Vector Machines	Training Time	Accuracy (%)	Details
Validation Fold 1	2h 30m	68.49	
Validation Fold 2	1h 54m	66.88	
Validation Fold 3	1h 55m	64.65	
Validation Fold 4	2h 12m	60.29	
Validation Fold 5	2h 48m	69.35	Each validation fold is divided into a
Validation Fold 6	1h 45m	63.23	training set (50%) and a test set (50%).
Validation Fold 7	3h 02m	58.41	
Validation Fold 8	1h 50m	67.16	
Validation Fold 9	2h 18m	61.21	
Validation Fold 10	1h 53m	64.33	
Average (Validation Folds)	2h 13	64.40	•
All Data	4h 05m	56.26	Training set: Randomly selected images (70%) Test set: Remaining images (30%).

Table 4.4. Evaluation of Support Vector Machines

Table 4.5. Evaluation of Gradient Boosting Machines

Gradient Boosting Machines	Training Time	Accuracy (%)	Details
Validation Fold 1	4h 11m	89.30	
Validation Fold 2	5h 06m	89.32	
Validation Fold 3	5h 40m	89.27	
Validation Fold 4	3h 59m	89.30	
Validation Fold 5	3h 31m	89.30	Each validation fold is divided into a
Validation Fold 6	4h 12m	89.32	training set (50%) and a test set (50%).
Validation Fold 7	5h 28m	89.24	
Validation Fold 8	5h 03m	89.30	
Validation Fold 9	5h 38m	89.29	
Validation Fold 10	3h 26m	89.34	
Average (Validation Folds)	4h 37m	89.30	
All Data	5h 10m	88.72	Training set: Randomly selected images (70%) Test set: Remaining images (30%)

Random Forest	Training Time	Accuracy (%)	Details
Validation Fold 1	623 s	98.00	
Validation Fold 2	554 s	97.96	
Validation Fold 3	550 s	97.94	
Validation Fold 4	641 s	98.02	
Validation Fold 5	629 s	97.98	Each validation fold is divided into a
Validation Fold 6	600 s	97.03	training set (50%) and a test set (50%).
Validation Fold 7	543 s	96.88	
Validation Fold 8	644 s	98.01	
Validation Fold 9	596 s	97.43	
Validation Fold 10	637 s	97.74	
Average (Validation Folds)	602 s	97.70	-
All Data	2012 s	96.37	Training set: Randomly selected images (70%) Test set: Remaining images (30%)

Table 4.6. Evaluation of Random Forest Algorithm

One of the main steps of XGBoost modeing is to tune the model parameters including the number of estimators in XGBoost, the learning rate, and maximum depth of the trees. Tables 4.7 and 4.8 present the results from XGBoost model.

In Table 4.7, 10 best models with different tuning parameters are shown. As presented in this table the best tuned model is when the number of estimators is 50, learning rate is 0.1 and the maximum depth is 3. The results in this table are the best result after grid search. The parameters that were involved in the grid search are as follows:

- n_estimators: [20, 50, 100, 200]
- Learning rate: [0.05, 0.1, 0.2, 0.3]
- max_depth:[2, 3, 4, 5, 6]
- Gamma: [0, 0.1, 0.2]

The other parameters such as Alpha and Lambda were used as default because they did not have

any significant effect on the performance of the model.

Table 4.8, shows the results of the best tuned model for validation and final training processes.

XGBoost	Parameters	Accuracy (%)	Details
Model 1	Estimators: 20, Learning Rate: 0.1, max(depth): 3	95.40	
Model 2	Estimators: 20, Learning Rate: 0.1, max(depth): 3	95.40	
Model 3	Estimators: 20, Learning Rate: 0.2, max(depth): 3	95.62	
Model 4	Estimators: 20, Learning Rate: 0.1, max(depth): 5	95.55	Training set:
Model 5	Estimators: 20, Learning Rate: 0.3, max(depth): 3	95.48	Randomly selected images
Model 6	Estimators: 50, Learning Rate: 0.1, max(depth): 3	95.72	Test set:
Model 7	Estimators: 50, Learning Rate: 0.2, max(depth): 3	95.29	Remaining images (30%)
Model 8	Estimators: 50, Learning Rate: 0.1, max(depth): 5	95.33	
Model 9	Estimators: 100, Learning Rate: 0.1, max(depth): 3	95.37	
Model 10	Estimators: 200, Learning Rate: 0.1, max(depth): 3	95.17	

Table 4.8. Evaluation	of XGBoost	Algorithm	after Tuning
-----------------------	------------	-----------	--------------

XGBoost (Tuned)	Training Time	Accuracy (%)	Details	
Validation Fold 1	113 s	96.30		
Validation Fold 2	106 s	96.32		
Validation Fold 3	54 s	96.23		
Validation Fold 4	85 s	96.28		
Validation Fold 5	93 s	96.30	Each validation fold is divided into a	
Validation Fold 6	96 s	96.32	training set (50%) and a test set (50%).	
Validation Fold 7	91 s	96.24		
Validation Fold 8	104 s	96.30		
Validation Fold 9	105 s	96.29		
Validation Fold 10	106 s	96.33		
Average (Validation Folds)	95.3 s	96.30		
All Data	125 s	95.72	Training set: Randomly selected images (70%) Test set: Remaining images (30%)	

Results using All Data	Training Time	Accuracy (%)
Logistic Regression	47.2 s	88.75
Support Vector Machines	2h 13m	64.40
Gradient Boosting Machines	4h 37m	89.30
Random Forest	602 s	97.70
XGBoost	98.2 s	93.14

Table 4.9. Evaluation Summaries

 Table 4.10. Comparison Between Different Algorithms in Optic Cup Detection

Results using All Data	Training Time	Accuracy (%)
Logistic Regression	115 s	88.51
Support Vector Machines	4h 05m	56.26
Gradient Boosting Machines	5h 10m	88.72
Random Forest	2012 s	96.37
XGBoost (Default)	154.7 s	93.14
XGBoost (Tuned)	125 s	95.72

The average training time and accuracy of all methods in the evaluation process are shown in Table 4.9; and the overall result of OC classification based on different models are summarized in

Table 4.10. As presentd in these tables, the performance of models are as follows:

Accuracy: Random Forest > XGBoost >> GBM >> Logistic Regression > SVM

Speed: Logistic Regression > XGBoost >> Random Forest >> SVM > GBM

The results show that the accuracy of Random Forest and XGBoost are very close to each other: 96.37% vs 95.72%, respectively. This table also indicates that the accuracy of Random Forest and XGBoost are significantly higher than the other mothods.

XGBoost has the training time of 125 seconds, but Random Forest needs 2012 seconds for training.

Thus, XGBoost is about 16 times faster than Random Forest.

Considering the evaluation results from Tables 4.9 and the overall results from Table 4.10, it is a strong conclusion that XGBoost outperforms all the other models in OC classification in terms of both accuracy and speed.

Chapter 5

Conclusions and Future Work

5.1. Conclusions

As the first main goal of this thesis, an image processing algorithm was developed to segment OD in fundus images. This algorithm is based on a combination of CLAHE, ASF, thresholding and CHT algorithms. The results of the proposed algorithm over a set of images from MESSIDOR database indicates that this method is highly precise (92.4%) in OD detection.

The second goal of the project was to introduce a set of features from the detected OD region to predict the OC and non-OC pixels. This goal is achieved by intruducing 24 features extracted for each pixel in OD region based on its actual and normalized distances from OD center, its level of red, green, blue and grey intensities, and features from CLAHE, Gabor , and ASF algorithms.

The extracted features were then used to implement predictive models to classify OD pixels into OC and non-OC. As the next goal of this project, three ensembled models were implemented: Random Forest, GBM and XGBoost for the classification purposes. These methods were then evaluated by comparing their results with the results from Logistic Regression and SVM methodologies.

The main contributions of this thesis research work are summarized as:

1- Development of an algorithm to precisely detect OD,

- 2- Generating predictive features for the purpose of OC classification,
- 3- Development of Random Forest, GBM, and XGBoost ensemble models to classify OD pixels into OC and non-OC pixels,
- 4- Well tuning XGBoost for high performance classification of OD with significantly high accuracy and fast training process.

5.2. Future Work

Since the results of XGBoost algorithm are extremely promising in optic cup detection, the proposed future work is mainly focused on this topic:

- 1- Implementation of XGBoost in Hadoop or Spark distributed systems to be trained with other retinal fundus image databases,
- 2- Developing feature extraction methods to define features on fundus image pixels for the purpose of labeling the images into OD, non-OD, and OC pixels in one step,
- 3- Implementation of a self-tuning XGBoost model with the capabilities to be tuned with different databases authomatically.

References:

- Z. Zhang *et al.*, "ORIGA-light: An online retinal fundus image database for glaucoma analysis and research," *Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBC'10*, pp. 3065–3068, 2010.
- M. D. Abràmoff, M. K. Garvin, and M. Sonka, "Retinal Imaging and Image Analysis," *IEEE Rev. Biomed. Eng.*, vol. 1, no. 3, pp. 169–208, 2010.
- [3] http://www.kumc.edu/school-of-medicine/ophthalmology/patient-care/retina-andvitreous.html.
- [4] M. D. T.; Rockaway, A Closer Look at the Eye : Researchers Develop New Retinal Imaging Technique. 2017, pp. 1–3.
- [5] M. Lalonde, M. Beaulieu, and L. Gagnon, "Fast and robust optic disc detection using pyramidal decomposition and hausdorff-based template matching," *IEEE Trans. Med. Imaging*, vol. 20, no. 11, pp. 1193–1200, 2001.
- [6] R. G. Ramani, S. Sugirtharani, and B. Lakshmi, "Automatic Detection of Glaucoma in Retinal Fundus Images through Image Processing and Data Mining Techniques," *Int. J. Comput. Appl.*, vol. 166, no. 8, pp. 38–43, 2017.
- [7] E. D. Cole, E. A. Novais, R. N. Louzada, and N. K. Waheed, "Contemporary retinal imaging techniques in diabetic retinopathy: a review," *Clin. Exp. Ophthalmol.*, vol. 44, no. 4, pp. 289–299, 2016.
- [8] K. K. W. Chan, F. Tang, C. C. Y. Tham, A. L. Young, and C. Y. Cheung, "Retinal

vasculature in glaucoma: a review," BMJ Open Ophthalmol., vol. 1, pp. 1–14, 2017.

- [9] M. Esmaeili, H. Rabbani, A. M. Dehnavi, and A. Dehghani, "Automatic detection of exudates and optic disk in retinal images using curvelet transform," *IET Image Process.*, vol. 6, no. 7, pp. 1005–1013, 2012.
- T. E. de Carlo, A. Romano, N. K. Waheed, and J. S. Duker, "A review of optical coherence tomography angiography (OCTA)," *IInternational J. Retin. Vitr.*, vol. 1, no. 5, pp. 1–15, 2015.
- [11] J. Nayak, R. Acharya U., P. S. Bhat, N. Shetty, and T. C. Lim, "Automated diagnosis of glaucoma using digital fundus images," *J. Med. Syst.*, vol. 33, no. 5, pp. 337–346, 2009.
- [12] E. Trucco *et al.*, "Validating Retinal Fundus Image Analysis Algorithms: Issues and a ProposalValidating Retinal Fundus Image Analysis Algorithms," *Invest. Ophthalmol. Vis. Sci.*, vol. 54, no. 5, pp. 3546–3559, 2013.
- [13] D. Vijayasekar, S. Dhivya, S. Dhanalakshmi, and S. Karthik, "Survey on Detection of Glaucoma in Fundus Image by Segmentation and Classification," *Int. J. Eng. Res. Technol.*, vol. 4, no. 09, pp. 529–532, 2015.
- [14] A. Budai, R. Bock, A. Maier, J. Hornegger, and G. Michelson, "Robust vessel segmentation in fundus images," *Int. J. Biomed. Imaging*, vol. 2013, pp. 1–22, 2013.
- [15] C. Sinthanayothin, J. F. Boyce, H. L. Cook, and T. H. Williamson, "Automated localisation of the optic disc, fovea and retinal blood vessels from digital color fundus images," *Br. J. Ophthalmol.*, vol. 4, no. 83, pp. 902–910, 1999.
- [16] T.-Y. Hsiao, C.-L. Wang, C.-N. Chen, F.-J. Hsieh, and Y.-W. Shau, "Elasticity of human vocal folds measured in vivo using color Doppler imaging.," *Ultrasound Med. Biol.*, vol. 28, no. 9, pp. 1145–1152, 2002.

- [17] K. Akyol, B. Şen, and Ş. Bayir, "Automatic Detection of Optic Disc in Retinal Image by Using Keypoint Detection, Texture Analysis, and Visual Dictionary Techniques," *Comput. Math. Methods Med.*, vol. 2016, pp. 1–10, 2016.
- [18] M. C. Viola Stella Mary, E. B. Rajsingh, and G. R. Naik, "Retinal Fundus Image Analysis for Diagnosis of Glaucoma: A Comprehensive Survey," *IEEE Access*, vol. 4, pp. 4327– 4354, 2016.
- [19] E. Decencière *et al.*, "Feedback on a publicly distributed image database: The Messidor database," *Image Anal. Stereol.*, vol. 33, no. 3, pp. 231–234, 2014.
- [20] R. Sahebrao Maher, S. N. Kayte, Sdd. College Bhokardan Babasaheb Ambedkar Marathwada, and M. Dhopeshwarkar, "Automated Diagnosis Non-proliferative Diabetic Retinopathy in Fundus Images using Support Vector Machine Sandip T. Meldhe," *Int. J. Comput. Appl.*, vol. 125, no. 15, pp. 975–8887, 2015.
- [21] K. Ghasemi Falavarjani, I. Tsui, and S. R. Sadda, "Ultra-wide-field imaging in diabetic retinopathy," *Vision Res.*, vol. 139, pp. 187–190, 2017.
- [22] Y. Zhang, Z. Ye, M. Wang, and N. Qiao, "Ganglion cell complex loss precedes retinal nerve fiber layer thinning in patients with pituitary adenoma," *J. Clin. Neurosci.*, vol. 43, pp. 274– 277, 2017.
- [23] S. G. Thorat, "Automated Glaucoma Screening using CDR from 2D Fundus Images," Int.
 J. Comput. Eng. Res., vol. 04, no. 5, pp. 2250–3005, 2014.
- [24] A. Almazroa, R. Burman, K. Raahemifar, and V. Lakshminarayanan, "Optic Disc and Optic Cup Segmentation Methodologies for Glaucoma Image Detection: A Survey," J. Ophthalmol., no. 1–28, 2015.
- [25] Z. Zhang, J. Liu, F. Yin, B. H. Lee, D. W. K. Wong, and K. R. Sung, "ACHIKO-K: Database

of fundus images from glaucoma patients," *IEEE Conf. Proc. Ind. Electron. Appl.*, pp. 228–231, 2013.

- [26] R. Bernardes, P. Serranho, and C. Lobo, "Digital ocular fundus imaging: A review," *Ophthalmologica*, vol. 226, no. 4, pp. 161–181, 2011.
- [27] R. P. Rajaiah and R. J. Britto, "Optic Disc Boundary Detection and Cup Segmentation for Prediction of Glaucoma," *Int. J. Sci. Eng. Technol. Res.*, vol. 3, no. 10, pp. 2665–2672, 2014.
- [28] T. Devasia, P. Jacob, and T. Thomas, "Automatic Optic Disc Boundary Extraction from Color Fundus Images," *Int. J. Adv. Comput. Sci. Appl.*, vol. 5, no. 7, pp. 117–124, 2014.
- [29] F. Fumero, S. Alayón, J. L. Sanchez, J. Sigut, and M. Gonzalez-Hernandez, "RIM-ONE : An open retinal image database for optic nerve evaluation RIM-ONE : An Open Retinal Image Database for Optic Nerve Evaluation," in *International Symposium on Computer-Based Medical Systems (CBMS)*, 2011, pp. 1–6.
- [30] J. Lowell *et al.*, "Optic Nerve Head Segmentation," *IEEE Trans. Med. Imaging*, vol. 23, no. 2, pp. 256–264, 2004.
- [31] P. C. Siddalingaswamy and G. K. Prabhu ., "Automatic Localization and Boundary Detection of Optic Disc Using Implicit Active Contours," *Int. J. Comput. Appl.*, vol. 1, no. 7, pp. 1–5, 2010.
- [32] R. J. Qureshi, L. Kovacs, B. Harangi, B. Nagy, T. Peto, and A. Hajdu, "Combining algorithms for automatic detection of optic disc and macula in fundus images," *Comput. Vis. Image Underst.*, vol. 116, no. 1, pp. 138–145, 2012.
- [33] D. A. Godse and D. S. Bormane, "Automated Localization of Optic Disc in Retinal Images," *Int. J. Adv. Comput. Sci. Appl.*, vol. 4, no. 2, pp. 65–71, 2013.

- [34] W. W. K. Damon, J. Liu, T. N. Meng, Y. Fengshou, and W. T. Yin, "Automatic detection of the optic cup using vessel kinking in digital retinal fundus images," *Proc. - Int. Symp. Biomed. Imaging*, pp. 1647–1650, 2012.
- [35] R. Ingle and P. Mishra, "Cup Segmentation by Gradient Method for the Assessment of Glaucoma from Retinal Image," *Int. J. Eng. Trends Technol.*, vol. 4, no. 6, pp. 2540–2543, 2013.
- [36] F. Yin *et al.*, "Automated segmentation of optic disc and optic cup in fundus images for glaucoma diagnosis," *Proc. - IEEE Symp. Comput. Med. Syst.*, 2012.
- [37] M. Mishra, M. K. Nath, and S. Dandapat, "Glaucoma Detection from Color Fundus Images," *Int. J. Comput. Commun. Technol.*, vol. 2, no. Vi, pp. 7–10, 2011.
- [38] T. Devasia, P. Jacob, and T. Thomas, "Automatic Optic Disc Localization and Segmentation using Swarm Intelligence," World Comput. Sci. Inf. Technol. J., vol. 5, no. 6, pp. 92–97, 2015.
- [39] K. Sakthivel and R. Narayanan, "An automated detection of glaucoma using histogram," *Int. J. Ophthalmol.*, vol. 8, no. 1, pp. 194–200, 2015.
- [40] E. R. Dougherty and R. a Lotufo, *Hands-on Morphological Image Processing*. Bellingham: SPIE PRESS, 2003.
- [41] Y. Chang, C. Jung, P. Ke, H. Song, and J. Hwang, "Automatic Contrast-Limited Adaptive Histogram Equalization with Dual Gamma Correction," *IEEE Access*, vol. 6, pp. 11782– 11792, 2018.
- [42] S. M. Pizer *et al.*, "Adaptive Histogram Equalization and Its Variations," *Comput. Vision, Graph. Image Process.*, vol. 39, pp. 355–368, 1987.
- [43] D. Ioannou, W. Huda, and A. F. Laine, "Circle recognition through a 2D Hough Transform

and radius histogramming," Image Vis. Comput., vol. 17, pp. 15-26, 1999.

- [44] J. Illingworth and J. Kittler, "The Adaptive Hough Transform," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-9, no. 5, pp. 690–698, 1987.
- [45] M. Merlin and B. Priestly Shan, "Robust and efficient segmentation of blood vessel in retinal images using gray-level textures features and fuzzy SVM," *Biomed. Pharmacol. J.*, vol. 8, no. 2, pp. 1111–1120, 2015.
- [46] H. F. Yu, F. L. Huang, and C. J. Lin, "Dual coordinate descent methods for logistic regression and maximum entropy models," *Mach. Learn.*, vol. 85, no. 1–2, pp. 41–75, 2011.
- [47] C. Campbell and Y. Ying, *Learning with Support Vector Machines*, vol. 5, no. 1. 2011.
- [48] A. Kadiyala and A. Kumar, "Applications of python to evaluate the performance of decision tree-based boosting algorithms," *Environ. Prog. Sustain. Energy*, vol. 37, no. 2, pp. 618–623, 2018.
- [49] A. Paul, D. P. Mukherjee, P. Das, A. Gangopadhyay, A. R. Chintha, and S. Kundu,
 "Improved Random Forest for Classification," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 4012–4024, 2018.
- [50] L. Breiman, "Random forests," Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001.
- [51] L. Breiman, "Population Theory for Boosting Ensembles," vol. 32, no. 1, pp. 1–11, 2004.
- [52] D. Mahapatra, "Analyzing Training Information From Random Forests for Improved Image Segmentation," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1504–1512, 2014.
- [53] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," *Front. Neurorobot.*, vol. 7, pp. 1–21, 2013.
- [54] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. 2008.
- [55] T. K. Ho, "The random subspace method for constructing decision forests," IEEE Trans.

Pattern Anal. Mach. Intell., vol. 20, no. 8, pp. 832-844, 1998.

- [56] J. H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine." pp. 1– 39, 2001.
- [57] L. Torlay, M. Perrone-Bertolotti, E. Thomas, and M. Baciu, "Machine learning–XGBoost analysis of language networks to classify patients with epilepsy," *Brain Informatics*, vol. 4, no. 3, pp. 159–169, 2017.
- [58] D. Nielsen, "Tree Boosting With XGBoost Why Does XGBoost Win 'Every' Machine Learning Competition?," 2016.
- [59] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," 2016.
- [60] B. Hofner, A. Mayr, N. Robinzonov, and M. Schmid, *Model-based boosting in R: A handson tutorial using the R package mboost*, vol. 29, no. 1–2. 2014.
- [61] M. Nishio *et al.*, "Computer-aided diagnosis of lung nodule using gradient tree boosting and Bayesian optimization," *PLoS One*, vol. 13, no. 4, pp. 1–14, 2018.
- [62] M. van Steen and A. S. Tanenbaum, "A brief introduction to distributed systems," *Computing*, vol. 98, no. 10, pp. 967–1009, 2016.
- [63] https://spark.apache.org/docs/2.3.1/.
- [64] https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html.
- [65] V. Kumar and N. Sinha, "Automatic Optic Disc segmentation using maximum intensity variation," in *IEEE 2013 Tencon Spring*, 2013, pp. 29–33.
- [66] I. U. Morphological, E. Detection, A. Aquino, M. E. Gegúndez-arias, and D. Marín,
 "Detecting the Optic Disc Boundary in Digital Fundus Images Using Morphological, Edge
 Detection, and Feature Extraction Techniques," *IEEE Trans. Med. Imaging*, vol. 29, no. 11,
 pp. 1860–1869, 2010.

- [67] S. Lu, "Accurate and efficient optic disc detection and segmentation by a circular transformation," *IEEE Trans. Med. Imaging*, vol. 30, no. 12, pp. 2126–2133, 2011.
- [68] A. Almazroa, "A novel Automatic Optic Disc and Cup Image Segmentation System for Diagnosing Glaucoma using RIGA dataset by," University of Waterloo, 2016.