2022-08

Detecting Glaucoma from Optic Fundus Image Using Machine Learning Approach

Kalkidan, Mulatu

BAHIR DAR UNIVERSITY

BAHIR DAR INSTITUTE OF TECHNOLOGY

SCHOOL OF RESEARCH AND POSTGRADUATE STUDIES

OF ELECTRICAL AND COMPUTER ENGINEERING

Master of Science in Computer Engineering

Detecting Glaucoma from Optic Fundus Image Using Machine Learning Approach

By:

Kalkidan Mulatu

Advisor: Getachew Alemu (PhD)

Co-Advisor: Atirsaw Aweke (MSc)

Bahir Dar, Ethiopia
August, 2022

Detecting Glaucoma from Optic Fundus Image Using Machine Learning Approach

By:
Kalkidan Mulatu

A thesis submitted to the school of Research and Graduate Studies of Bahir Dar Institute of Technology, BDU in partial fulfillment of the requirements for the degree of
MSc in the Computer Engineering in Faculty of Electrical and Computer Engineering.

Advisor: Getachew Alemu (PhD)

Co-Advisor: Atirsaw Aweke (MSc)

Bahir Dar, Ethiopia
August, 2022

Declaration

I, the undersigned, declare that the thesis comprises my work. In compliance with internationally accepted practices, I have acknowledged and refereed all materials used in this work. I understand that non-adherence to the principles of academic honesty and integrity, misrepresentation/ fabrication of any idea/data/fact/source will constitute sufficient ground for disciplinary action by the University and can also evoke penal action from the sources which have not been properly cited or acknowledged.

Name of the student: ___Kalkidan Mulatu______ Signature: ___________________

Date of submission: ___________________________

Place: Bahir Dar

This thesis has been submitted for examination with my approval as a university advisor.

Advisor: Dr. Getachew Alemu

Advisor's Signature: Getachew Alemu. ____________________

BAHIR DAR UNIVERSITY

BAHIR DAR INSTITUTE OF TECHNOLOGY

SCHOOL OF GRADUATE STUDIES

Electrical and Computer Engineering

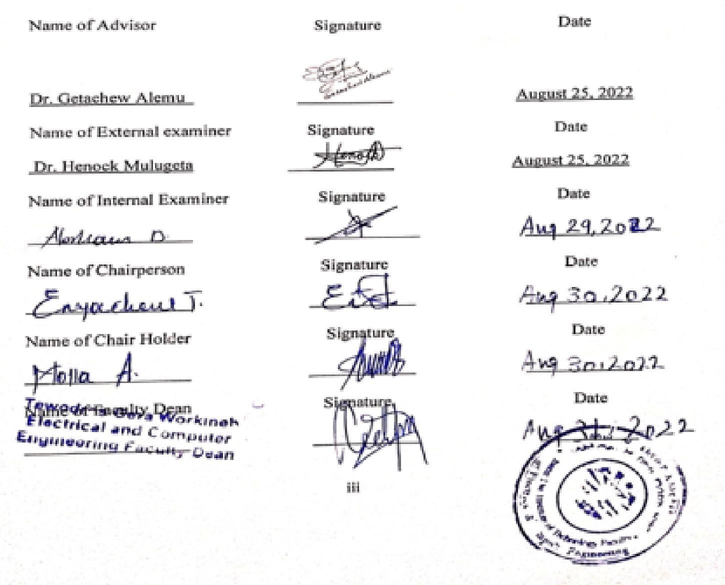
Approval of thesis for defense result

I hereby confirm that the changes required by the examiners have been carried out and incorporated in the final thesis.

Name of Student Kalkidan Mulatu Signature ________________ Date ____________

As members of the board of examiners, we examined this thesis entitled "Detecting Glaucoma from Optic Fundus Image Using Machine Learning Approach" by *Kalkidan Mulatu*. We hereby certify that the thesis is accepted for fulfilling the requirements for the award of the degree of Masters of science in "Computer Engineering".

Board of Examiners



Name of Advisor	Signature	Date
Dr. Getachew Alemu		August 25, 2022
Name of External examiner	Signature	Date
Dr. Henock Mulugeta		August 25, 2022
Name of Internal Examiner	Signature	Date
Abrham D.		Aug 29, 2022
Name of Chairperson	Signature	Date
Enyachew T.		Aug 30, 2022
Name of Chair Holder	Signature	Date
Molla A.		Aug 30, 2022
Name of Faculty Dean	Signature	Date
Tewodros Gera Workineh Electrical and Computer Engineering Faculty Dean		Aug 31, 2022

Acknowledgments

First and foremost, I would like to praise and thank God, the Almighty, and his beloved mother holy Mary. For giving me strength and encouragement throughout all the challenging moments of completing this thesis work. I am truly grateful for their unconditional and endless love, mercy, and grace.

Words cannot express my gratitude to my advisor Dr. Getachew Alemu, this research work would not have been possible without his exceptional support, and also am thankful for inspiring me and keeping my work on track from my first encounter up to now. Also, I would like to express my deepest gratitude to my co-advisor Atirsaw Aweke.

I am also grateful to my family, especially my parents (Elsa Abriham and Mulatu Wudeneh) and my brother (Ermias Mulatu), for their help and moral support. Their belief in me has kept my spirits and motivation high during this process.

I must express my gratitude to a very special person, my husband, Abenezer Dechasa(kiya), for his unwavering love, support, and understanding throughout my pursuit of an MSc degree, which enabled me to finish this thesis. Kiyashe, I love you.

Finally, I would be remiss in not mentioning my friends Agere Birihanu and Misirak Mekango, who went through hard times together, cheered me on, and celebrated each accomplishment with me. Last but not least, I am thankful to my cats for all the entertainment and emotional support they gave me during this stressful moment.

Abstract

Glaucoma is an eye condition that occurs as the result of increased fluid pressure inside the eye and injured optic nerve when intraocular pressure rises. If left untreated, this can permanently damage vision in the affected eye(s) and result in blindness. It cannot be cured; however, its progression can be slowed down by treatment. Therefore, early glaucoma detection is crucial. The available scanning methods for glaucoma approaches are pricy and require the assistance of qualified medical professionals and also the manual examination of the eye takes a lot of time, and different doctors' parameter measurements may not always be accurate. Because of this, it's critical to develop a machine learning method that can quickly and easily identify the disease to assist medical professionals or different non-expert users. We created a CNN model to automatically assess fundus images of the eye to determine if a person has glaucoma or not. Many studies have been conducted by utilizing machine learning to diagnose glaucoma, but the methods they employed for preprocessing and feature extraction are still limited the accuracy of the detection. To reduce this limitation an alternative glaucoma detection model is developed based on a machine learning method to analyze and classify the fundus image. The proposed system is implemented using the Python programming language on top of the Tensor flow and Keras API and it is tested with publicly available Large-scale Attention-based Glaucoma (LAG) dataset. Preprocessing methods like Bicubic interpolation, noise reduction with a median filter, and contrast enhancement with contrast limited adaptive histogram equalization (CLAHE) were used to speed up processing and improve the accuracy of the detection model. Additionally, after preprocessing, relevant features from fundus images were extracted using buildup CNN, GLCM and CNN-GLCM to compare the result. The obtained best features were then given to the SVM KNN and CNN classifiers. Our model is evaluated using performance metrics like accuracy, precision, recall, and f1-score. From the experiment, we noticed that the developed end-to-end CNN with adam optimizer achieved a significant classification performance using preprocessed images with training accuracy of 98.40% and test accuracy of 93% with a batch size of 32 and 100 epoch.

Keywords: Glaucoma, Machine Learning, Preprocessing, CNN, GLCM, KNN

Table of Contents

List of abbreviations

Adagrad	Adaptive Gradient
Adamax	Adaptive Max Pooling
Adam	Adaptive Momentum
AHE	Adaptive histogram equalization
AMD	Age-related macular degeneration
ANN	Artificial neural networks
AUC	Area Under the ROC Curve
BM3D	Block Match 3D
CDR	Cup to Disc Ratio
CGSA	Chinese Glaucoma Study Alliance
CLAHE	Contrast limited adaptive histogram equalization
CNN	Convolutional Neural Network
CV	Computer Vision
DR	Diabetic Retinopathy
FC	Fully connected
GLCM	Gray Level Co-occurrence Matrix
HDTV	High-Definition Television
HE	Histogram equalization
HRF	High-Resolution Fundus
HSV	Hue, Saturation, And Value
ICA	Independent Component Analysis
IDE	Integrated Development Environment
IOP	Neuro-retinal rim
KNN	K-Nearest Neighbor
LAG	Large-scale Attention-based Glaucoma
LDA	Linear Discriminant Analysis
ML	Machine learning
MMD	Minimum Mean Distance

Nadam	Nesterov-accelerated Adaptive Moment Estimation
NN	Neural Network
OCT	Optical Coherence Topography
OD	Optical Disk
ONH	Optical Nerve head
PCA	Principal Component Analysis
PSNR	Peak Signal-to-Noise Ratio
RBF	Radial Basis Function
ReLU	Rectified Linear Unit
ResNET	Resdual Neural Network
RF	Random Forest
RGB	Red Green Blue
RGC	Retinal ganglion cells
RMSProp	Root Mean Square Propagation
RNFL	Retinal nerve fiber layer
ROC	Receiver Operating Characteristic
ROI	Region of Interest
SD	Standard Deviation
SGD	Stochastic gradient descent
SLO	Scanning Laser Ophthalmoscopy
SNR	Signal-to-Noise Ratio
SVM	Support vector machines
TN	True negative
TP	True positive
VGG	Visual Geometry Group
WHO	World Health Organization

List of Figures

List of Tables

CHAPTER ONE

INTRODUCTION

1.1 Background

The most powerful of the five senses, vision, is important in every aspect of our life. According to a report released by the World Health Organization (WHO) in 2019, at least 2.2 billion individuals worldwide have a vision impairment, with at least 1 billion of these having a vision impairment that might have been avoided or that has yet to be addressed. This 1 billion people includes those with mild to severe distance vision impairment or blindness due to untreated refractive error (123.7 million), cataract (94 million), glaucoma (7.7 million), corneal opacities (4.2 million), diabetic retinopathy (3.9 million), and trachoma (2 million), as well as those with near vision impairment due to uncorrected presbyopia (826 million). (WHO, 2019).

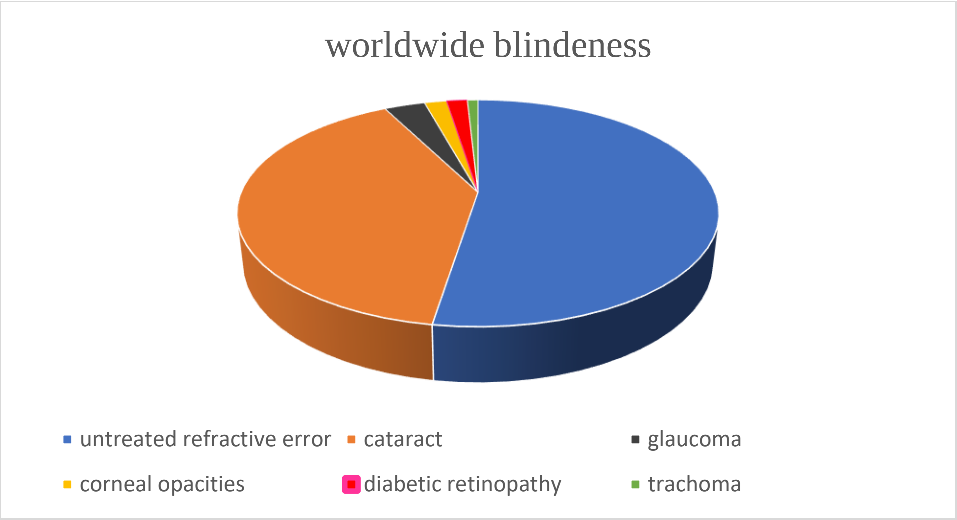


Figure 1.1: WHO report on causes of worldwide blindness

Glaucoma is a chronic and irreversible eye disease that is a neurodegenerative condition which is the progressive loss of the retinal ganglion cells (RGC) leading to visual loss. It results in varying degrees of irreversible visual disability and in some cases blindness. (Guidelines et al., 2017). Glaucoma is responsible for around 1.9 percent of global blindness, with a rate of around 15% in Africa. According to WHO, between 2020 (76 million) and 2030 (95.4 million), the number of people with age-related eye ailment glaucoma is expected to climb 1.3 times. When we come to Ethiopia it is the fifth leading cause of blindness and is responsible for 5.2 percent of blindness. (WHO, 2019)(Wubet & Assefa, 2021)

Glaucoma is characterized by the progressive degeneration of optic nerve fibers, which results in structural changes to the optic nerve head and nerve fiber layer, as well as visual field functional failure. (Kolb et al., 1995). As figure 1.1 shows the neuro-retinal rim (IOP) is the annular zone between the optic disc and the cup boundary that represents the retinal nerve fibers. Intraocular pressure refers to the fluid pressure within the eye's inner chamber. The blockage of the aqueous humor outflow is caused by an increase in this IOP. This causes damage to the optic nerve, which is responsible for transmitting information from the retina to the brain. The thickening of the retinal nerve fiber layer (RNFL) as a result of the degeneration of optic nerve fibers is known as 'cupping.' Glaucoma progresses as a result of this cupping. By measuring the cup to disc ratio (CDR), which is an indicator of glaucomatous change, the decrease in healthy neuro-retinal tissues can be easily seen. CDR for a healthy eye is usually about 0.3. (R. Bourne et al., 2012)(Law & Coleman, 2016). Hazy or blurred vision, the appearance of rainbow-colored circles around bright lights, acute eye and head pain, nausea or vomiting (together with significant eye pain), and sudden sight loss are all symptoms of glaucoma. (DOREEN FAZIO, n.d.)

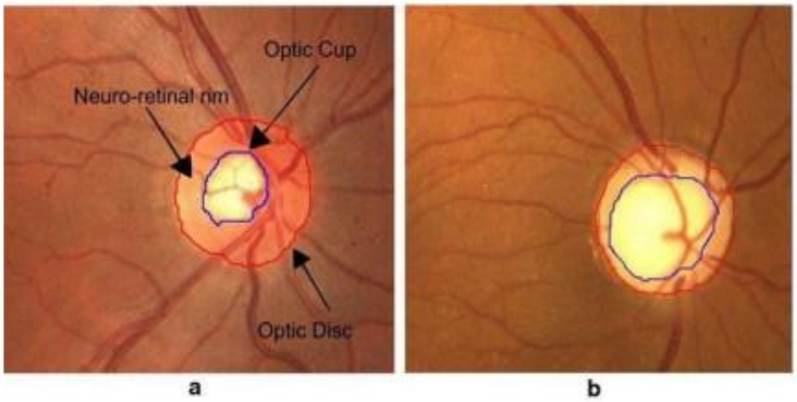


Figure 1.2: Normal and glaucomatous optic discs. at, “a” A normal disc with a physiological excavation surrounded by a healthy neuroretinal rim. “b”, A completely excavated disc with no neuro-retinal rim left as seen in end stage glaucoma.(Pinto et al., 2019)

1.1.1 A Comprehensive Glaucoma Diagnostic Exams

According to (Exam, 2014; National Glaucoma Research, 2021) before making a glaucoma diagnosis, five things should be checked: Two regular eye examinations are included in glaucoma check-ups: ophthalmoscopy and tonometry

Tonometry

Tonometry is a test that measures the pressure inside the eye. Eye drops are used to numb the eye during tonometry. The inner pressure of the eye is then measured using a tonometer by a doctor or technician. A little device or a heated puff of air is used to apply a modest amount of pressure on the eye. Normal pressure ranges from 12 to 22 mm Hg ("mm Hg" refers to millimeters of mercury, a scale used to measure eye pressure). When the pressure in the eye exceeds 20mm Hg, glaucoma is diagnosed. Some persons, however, might develop glaucoma at pressures ranging from 12 to 22 mm Hg. Each person's eye pressure is different.

Ophthalmoscopy

This diagnostic process allows the doctor to check for glaucoma damage in the optic nerve. Eye drops dilate the pupil, allowing the doctor to check the structure and color of the optic nerve through an eye. The doctor will next light and enlarge the optic nerve with a little instrument with a light on the end. If the IOP is abnormally high or the optic nerve appears abnormal, the doctor may order one or two further glaucoma exams: gonioscopy and perimetry.

Perimetry

Perimetry is a visual field test that generates a map of a whole visual field. This test will assist a doctor in determining whether or not the patient has glaucoma.

Gonioscopy

This test determines if the angle between the iris and the cornea is open and wide or closed and narrow. Then the doctor examines if the angle between the iris and the cornea is closed and blocked (a symptom of angle closure or acute glaucoma) or broad and open (a sign of open-angle, chronic glaucoma).

Pachymetry

Pachymetry is a painless test that determines the thickness of cornea, which is the transparent window in front of an eye. To measure the thickness of the cornea, a probe called a pachymeter is gently placed on the front of patient eye. The doctor can better comprehend the IOP reading and establish a treatment plan that is suited for the patient using this measurement.

Glaucoma Imaging Tests

A doctor can accurately track the progression of glaucoma with glaucoma imaging tests. The examinations are harmless and radiation-free. The doctor will use eye drops to dilate the pupils before utilizing a digital camera to take pictures of the patient's optic nerve or using other technologies Optical Coherence Topography (OCT) to map the patient's optic nerve. No matter the modality the doctor uses, the results are immediately available and can be displayed for discussion on a computer screen.

Utilizing the aforementioned procedures to make a clinical diagnosis of the eye takes time and involves inter and intra-observer variability. Here digital screening methods for the early diagnosis of glaucoma from fundus images are very beneficial since they make it easier to stop the disease's progression. Fundus images make it easy to see the interesting clinical characteristics of the eye, such as the retina, optic disc, blood vessels, etc. Additionally, as figure 1.3 shows fundus cameras can be used to assess a variety of structures, including changes in cup to disc ratio, the ONH, cup diameter, and more. The main benefit of retinal fundus images is that then they are generally simple to obtain in clinical settings. Then, finding the ocular disorders of glaucoma is done by extracting the data from the digital picture analysis. Age-related macular degeneration (AMD), glaucoma, and diabetic retinopathy (DR) can all be monitored for progression using digital fundus images taken with a fundus camera.(Vlachokosta et al., 2007)

Despite the fact that images of the glaucoma fundus reveal a complicated mixture of hidden visual patterns, only an ophthalmologist with the requisite knowledge and expertise can see and examine these patterns and the examination also took a lot of time. Therefore, computer-aided technologies like machine learning play an important role in order to give supervision for examination and assessment to simplify this difficulty of diagnosis. In this thesis, machine learning methods for glaucoma detection were developed using fundus images. And from those images, the developed model would extract various features, and the classifier would use these features as input.

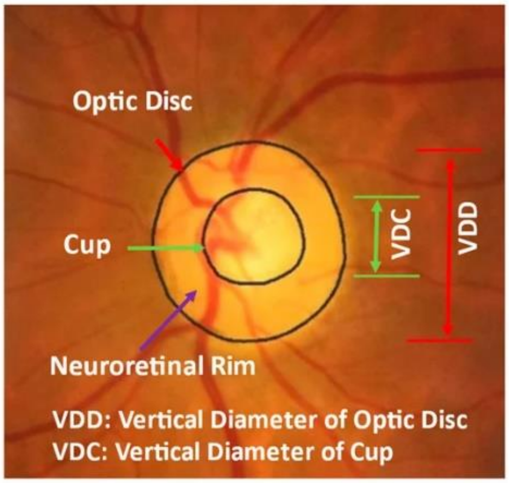

Optic Disc
Cup
VDC
VDD
Neuroretinal Rim
VDD: Vertical Diameter of Optic Disc
VDC: Vertical Diameter of Cup

Figure 1.3: Optic disc, cup, and neuro-retinal rim in a fundus image.(R. R. A. Bourne & Khatib, 2021)

1.2. Statement of the problem

Glaucoma is a chronic, incurable eye condition that impairs vision and is one of the main causes of blindness. It is a devastating chronic eye condition in which the optic nerve gradually undergoes damage. It is also known as the “silent thief” of sight since symptoms don't show up until the condition is fairly advanced. Even though glaucoma cannot be cured, therapy can halt its progression. Therefore, early glaucoma detection will be essential for a quick diagnosis. Effective imaging is crucial for the early identification of glaucoma. Currently one of the most common and important methods for diagnosing glaucoma is by using digital fundus images. One of the main physiological parameters for the diagnosis of glaucoma is the CDR in retinal fundus images, which serves as a significant indicator of whether the observer has glaucoma or not.

Machine learning methods are used by various researchers, but in the present state of the art, these algorithms when combined with deep learning can increase the model's accuracy level. This research study intends to develop glaucoma early detection methods that are more efficient and effective without the need for an experienced ophthalmologist. In this study, deep neural networks are used to learn high-level features incrementally from digital fundus images, reducing or simplifying the job of domain experts and avoiding strict feature extraction. Therefore, creating a system utilizing machine learning will assist physicians and ophthalmologists in more swiftly and effectively detecting, diagnosing, and treating the condition. The results of this study are anticipated to enhance glaucoma identification.

Finally, this thesis will answer the following research questions to attain the specific objectives.

- What effective image-processing technique should be applied to improve the fundus image?
- To what extent preprocessing increase the performance of CNN model?
- Which optimizer achieve the best in our model and dataset?
- To what extent the identification accuracy is registered?

1.3. Objectives of the study

1.3.1. General objective

The general objective of this research is to develop and detect glaucoma from digital optic fundus images using machine learning approaches.

1.3.2. Specific objectives

In order to achieve the general objective, the following specific objectives are identified.

- To collect glaucoma fundus image datasets.
- To study and select preprocessing technique and classification algorithm
- To investigate the impact of preprocessing methods like bicubic interpolation, median filtering and CLAHE in the glaucoma detection
- Investigating the effect of CNN and GLCM feature extraction techniques
- To design and develop an enhanced glaucoma detection model from fundus image.
- To evaluate the performance of the proposed model with the best performance metrics.

1.4. Methodology

The methods, tools, and procedures listed below are employed throughout the research activity to accomplish the aforementioned study objectives and are described in more detail in the following sections.

1.4.1. Literature review

The literature on medical image processing and machine learning-related technologies for early detection of glaucoma based on various techniques, preprocessing techniques, and feature extraction techniques that help to increase the accuracy of glaucoma detection are reviewed in order to achieve the above-mentioned research objectives.

1.4.2. Image acquisition and dataset preparation

This stage is essential since the quality and quantity of data collected will have a direct impact on how accurate the predictive model can be. Therefore, the most recent and accessible site, the Large-scale Attention-based Glaucoma (LAG) database, was used to gather the digital optical fundus image collection.

1.4.3. Data pre-processing

Several pre-processing techniques were used after the dataset was gathered. Since the data was collected in varied sizes, it needed to be standardized, and data pixel equalization was done. The effect of the proposed model was investigated using the Bi cubic interpolation method for resizing the fundus image. Therefore, in order to simplify computation and improve the suggested model's accuracy, we carry out various pre-processing techniques such as histogram equalization, noise removal, image standardization, and data augmentation. Image datasets are split into two categories: training datasets used to develop the model and boost system performance through various parameters, and testing datasets used to assess the system. It will be divided into training validating and testing in an 80/10/10 format.

1.4.4. Feature extraction and model training

Following the pre-processing phase, features were extracted and classified using a different classification method to select the best one. Following that, various feature extraction and classification methods were evaluated, and for an end-to-end model, various optimizers according to various learning rates were evaluated.

1.4.5. Performance evaluation technique

The final performance or accuracy of the developed model was assessed in this study using the f1 score value, precision, recall value, accuracy, and confusion matrix. The confusion matrix, which is intended to assess the effectiveness of the classification models for a specific set of test data, was used to compare the system's output against the observed data. Based on the input, or training data, accuracy was used to measurement the performance of the model.

1.4.6. Implementation tool

An implementation tool for this study was Python running on the Windows operating system. Python is a fantastic high-level programming language with an interactive environment for machine learning and computer vision because it is very flexible, independent of platforms, and used by millions of engineers and scientists worldwide. As machine learning (ML) requires continuous and bunch data processing, Python's built-in libraries enable us to access, manipulate, and transform data in the most practical and efficient ways possible.

1.5. Scope and limitation

1.5.1. Scope

The main focus of this research is designing, modeling, and prototype development of automatic detection glaucoma by different machine learning approach. However, this research work does not include the recommendation of treatment at each stage or severity level of glaucoma.

1.5.2. Limitation

There are four different types of glaucoma: primary open-angle, angle-closure, normal tension, and secondary. Our research has some limitations, such as the fact that it is not real-time and that our model only determines whether a patient has glaucoma or not, not grade the type of glaucoma they have. Another limitation is that computational costs vary depending on the quality of the tools used to compile the image dataset as well as the machines used to implement the algorithms and the classifiers employed, in addition to the parameters and parameters used.

1.6. Significance of the study

The Ethiopian government has made every effort over the past few years to combat blindness, which it recognizes as one of the country's significant health issues.(WSPOS ethiopia, n.d.). According to the report from (Ministry, 2003) on the national strategic action plan for eye health 2016 - 2020 from the major causes of blindness in Ethiopia, the Prevalence rate of glaucoma was 5.2 % of the total population. It is estimated based on the National Survey of 2006 and the projected population of Ethiopia in 2014. But by making a diagnosis as soon as possible, we can minimize this outcome since glaucoma's progression can be diagnosed if we detected it early. However, it is challenging to do manual glaucoma screening for all suspected individuals due to the shortage of experienced ophthalmologists and the examination process takes time. There are 132 general ophthalmologists, including 20 subspecialists, working in the nation, according to reports from 2015. About 60% of the ophthalmologists in practice are located in Addis Ababa, and 49% are employed in the private or non-profit sectors. Ethiopia has substantially fewer working professionals than the globally advised ratios of one ophthalmologist for every 250,000 people, one optometrist for every 50,000 people, and one cataract surgeon for every 250,000 people. (Ministry, 2003) Therefore, it is vital to design an autonomous glaucoma detection system that is quite accurate and effective.

The first and most important outcome of this research is that it gives ophthalmologists a lot of support in interpreting the results of digital fundus images to diagnose glaucoma. This study also speeds up the detection and diagnosis procedure, which saves patients' and ophthalmologists' time and effort. The other, the equally significant one, is reducing the need for ophthalmologists, particularly in rural areas. Giving rural residents who lack access to ophthalmologists the chance to quickly discover the health of their eyes. The second significance is that the performance of the suggested model was improved by using a better pre-processing technique.

1.7. Thesis Organization

This section presents an overview of the contents of the remaining chapters. The rest of this thesis is organized as follows.

In Chapter Two, general image processing techniques, feature extraction, and classification approaches that are used to enhance the performance of the model are discussed. At the end of the chapter brief review of literature on the detection of glaucoma by using fundus images was discussed in detail.

In Chapter Three, the proposed system is described in detail. The system design part shows the model of our work, and the proposed system to detect the specified problem. Also in this chapter, different performance evaluation techniques that are used to evaluate the proposed model are discussed in detail.

In Chapter Four, different experimental results and evaluation of the proposed model for detecting glaucoma are discussed. The response of the system for different optimizers, presence and absence of preprocessing and data augmentation is discussed.

In Chapter Five, the conclusion and recommendations are discussed based on the result of the study.

CHAPTER TWO

LITERATURE REVIEW

2.1. Overviews

The essential procedures for glaucoma detection are discussed in this chapter, including image resizing, noise reduction, image enhancement, feature extraction approaches and algorithms, and classifiers for pattern recognition or classification. In this section review of existing methodologies, sources of datasets for model training and testing, and concepts and functions of image classification are done. The final section of this chapter discusses other related work that has been conducted on glaucoma detection.

2.2. Retinal imaging

A variety of technologies, including cameras, scanners, and other equipment, can be used to collect images. However, because they cannot take images from the inside section of the body, using those imaging techniques for medical purposes is complicated especially when we come to the eye (retinal or optical structure of the eye). An ophthalmoscope is a device that doctors have traditionally used to examine the back of patients' eyes(Exam, 2014). Nowadays doctors can obtain a considerably larger digital view of the retina thanks to retinal imaging. A retinal camera is a tool used to take pictures of the retina, which is the internal surface of the eye. As (Abràmoff & Kay, 2012) mentions, in the past 160 years, retinal imaging has advanced quickly and is now an essential component of the clinical care and management of patients with both retinal and systemic illnesses. Fundus photography is frequently utilized for age-related macular degeneration, glaucoma, and diabetic retinopathy (DR) of population-based, large-scale detection.

(Abràmoff & Kay, 2012) defines fundus imaging as “the process whereby reflected light is used to obtain a two-dimensional (2D) representation of the 3D, semitransparent, retinal tissues projected on to the imaging plane”. The broad category of fundus imaging includes the following modalities/techniques:

Fundus photography (including so-called red-free photography): Image intensities show how much light from a particular waveband is reflected.

Color fundus photography: image intensities represent the amount of reflected red (R), green (G), and blue (B) wavebands, as determined by the spectral sensitivity of the sensor.

Stereo fundus photography: For depth resolution, image intensities show how much light is reflected from two or more different view angles.

Scanning Laser Ophthalmoscopy (SLO): Image intensities show how much reflected single-wavelength laser light was captured throughout time.

Adaptive optics SLO: Image intensities show how much-reflected laser light has been optically adjusted by simulating its wave front distortions.

2.3. Digital Image Processing

According to simple terms, image processing is the act of utilizing a digital computer to remove noise and other irregularities from digital images. Cheap and potent digital computers have grown widely accessible during the last four to five decades and are used for a variety of purposes. During those decades, a variety of image processing techniques were created. The main benefits of digital image processing techniques are their adaptability, repeatability, and ability to maintain the accuracy of original data. Image acquisition, image pre-processing, image enhancement, image segmentation, feature extraction, recognition, and classification are the core concerns in image processing. (Chitradevi & Srimanthi, 2014)

2.3.1. Image Acquisition

The initial stage in image processing is image acquisition; without it, the subsequent processes cannot be carried out. It is the process of taking a picture with equipment designed for photography. A database or other source specifically designed for research purposes may also be used to obtain images. The majority of the time, before feeding machine learning algorithms with an image from a database or one that we took ourselves, additional processing is required.

2.3.2. Image preprocessing

Before feeding the machine learning algorithm with image data, image pre-processing is a process to transform the image data. Although geometric transformations of images (such as rotation, scaling, and translation) are categorized as pre-processing techniques, the goal of pre-processing is an improvement of the image data that suppresses unintentional distortions or enhances some image features crucial for subsequent processing.(Vision, 1993)

2.3.2.1. Image resize

An important preprocessing step in computer vision is image resizing. For the purpose of showing visual content at various aspect ratios and resolutions, image resizing is crucial. With a better understanding of image semantics, image resizing can be done more successfully. Traditional methods of resizing images include scaling and cropping, both of which are straightforward and simple to use but have certain disadvantages. They perform changes without necessitating the preservation of semantic data, leading to obvious artifacts including content loss, boundary breaking, and oversqueeze.(Priyanka C. Dighe, 2014)

Interpolation is particularly important when picture resizing or resampling is done to satisfy the requirements of the transmission channel or to deliver the final image without any visual loss. Therefor the goal of image interpolation is to obtain the most accurate estimation of a pixel's intensity based on the values of its nearby pixels.(Han, 2013) Typically, one of three techniques is used to interpolate an image:

Nearest neighbor interpolation

It is the simplest interpolation. This technique assigns the value of the closest sample point in the input image to each interpolated output pixel. Even though this method is incredibly effective, the image quality is terrible because the gain in the pass band of a rectangular function's Fourier Transform, which is similar to a sinc function, quickly diminishes. Additionally, it has noticeable side lobes that are scaled logarithmically.(Fadnavis, 2014). The interpolation kernel for the nearest neighbor

$$h(x) = \begin{cases} 0 & |x|>0 \\ 1 & |x|<0 \end{cases} \qquad (2.1)$$

The frequency response of the nearest neighbor kernel is H(ω) = sinc (ω/2)

Although this method is very efficient, the quality of image is very poor. It is because the Fourier Transform of a rectangular function is equivalent to a sinc function; with its gain in pass band falls off quickly. Also, it has prominent side lobes are in the logarithmical scale.

Bi-cubic interpolation

In a two-dimensional regular grid, bicubic interpolation is better compared to cubic interpolation. The interpolated surface is smoother than the similar surfaces generated using the previously discussed methods of bilinear interpolation and nearest-neighbor interpolation. In bicubic interpolation polynomials, cubic, or cubic convolution algorithms are used. The weighted average of the 16 pixels that are closest to the specified input coordinates is used by the Cubic Convolution Interpolation to calculate the value of the grey level, which is then applied to the first four one-dimensional output coordinates. The number of grid points required to evaluate the interpolation function for bicubic interpolation (cubic convolution interpolation in two dimensions) is 16, with two grid points on either side of the point under consideration in both the horizontal and perpendicular directions.

2.3.2.2. Noise removal

The removal of noise from images is a significant issue. Noise is unwanted image contamination. Images are invariably contaminated by noise during acquisition, compression, and transmission due to the influence of the environment, transmission channel, and other variables. This results in distortion and loss of image information. Noise has a negative impact on potential post-image processing tasks like video processing, image analysis, and tracking. As a result, image de-noising is crucial in contemporary image processing systems. Image de-noising is the process of removing noise from a noisy image, so as to restore the true image. (PADZIL, 2016)(Fan et al., 2019). In the case of digital photographs, the most common noise types include Gaussian noise, Poisson noise, Speckle noise, Salt and Pepper noise, and many others. Researchers employ a variety of filters to eliminate noise. The wiener filter, bilateral filter, mean filter, median filter, and others are used to filter out the noise. Since filtering is a key component of image processing, several spatial filters, which may be further divided into two types: linear filters and non-linear filters, have been used for image de-noising. Filtering known as linear filtering occurs when the values of the pixels nearby the input pixel are combined linearly to get the value of the output pixel. Contrarily, nonlinear filters have an output that is not a linear function of their input. In comparison to non-linear approaches, linear methods are quicker, but they are less able to maintain the image's finer details.(Win et al., 2019).

I. Non- Linear filter

Median filter

A common example of a nonlinear filter used in image processing is the median filter. The median filter works by ordering the pixel values in a neighborhood, finding the median value from that ordered list, and then substituting that value for the original pixel in the output image. It solves the mean filter's inability to minimize impulsive noise, which is one of its drawbacks. The median filters are also helpful for maintaining a picture's edges while lowering image noise. (PADZIL, 2016). Gaussian, random, and salt-and-pepper noise can be eliminated while keeping the edges preserved using a median filter.

Anisotropic diffusion filter

In the context of image processing, an isotropic diffusion filter has been used successfully to eliminate high frequency noise while preserving the major edges of existent objects. It is a method of texture filtering that enhances the quality of textures when seen from various perspectives. This is most noticeable on flat, long-stretching walls or floor textures. Without texture filtering, a texture that is viewed at an angle loses a lot of clarity and blurs as the angle increases. As a result, they improve the quality of obliquely seen images. (Palma et al., 2014)

BM3D filter

By using block matching, related patches are piled into 3D groups in this technique, which then undergoes a wavelet domain transformation. Then, in the wavelet domain, hard thresholding or coefficient-based Wiener filtering is used. All estimated patches are then pooled to rebuild the entire image after an inverse transform of coefficients. But as the noise level gradually rises, BM3D's de-noising effectiveness significantly declines and artifacts appear, especially in flat areas. (Davob et al., 2007)

Bilateral filtering (BF)

Bilateral filtering (BF) is a non-iterative and local approach to edge-preserving smoothing. The intensity value of each pixel is replaced with an average value weighted by the geometric and photometric similarities between surrounding pixels within a spatial window to create a filtered image. (Wong & Chung, 2004).It only ceases to be iterative if a large spatial scope is applied (15 pixels in each dimension). However, a large spatial window could over smooth the image's sharp ridges and gutters.

2.3.2.3. Contrast enhancement

In various disciplines, including medical image analysis, remote sensing, high-definition television (HDTV), hyperspectral image processing, industrial X-ray image processing, microscopic imaging, etc., image enhancement (one of the significant approaches in digital image processing) plays a big role. Image enhancement is the processing of an image to make it better suited for a certain application. It is mostly used to enhance the image's clarity and visual effects, or to make the original image easier for computers to process. (He et al., 2011) A variety of approaches are used in the image enhancement process in order to increase an image's visual appeal or change it into a format that is more suitable for automated or human analysis. Here is a list of a few of them.

Histogram equalization (HE)

There are numerous techniques for improving visual contrast, and they can be broadly divided into two types: direct techniques and indirect techniques. Due to their clarity and simplicity, the histogram alteration approaches, of which histogram equalization (HE) is one of the most popular, have been used extensively among indirect methods.

Basically, HE involves translating each input pixel to its corresponding output pixel. In order to produce an improved output image, this method equalizes the intensity values to the histogram's full range. By boosting each pixel's value, it improves the contrast and brightness of the input image, leading to dynamic range expansion. One issue with histogram equalization is that it alters an image's brightness, making it unsuitable for consumer electronics devices where maintaining the original brightness and boosting contrast are crucial to prevent obtrusive distortions. (Kotkar, 2013).(Patel et al., 2020)

Adaptive histogram equalization (AHE)

A modified version of the histogram equalization approach is the adaptive histogram equalization method. This technique applies enhancement processes to a selected area of any image and adjusts contrast based on the pixels around it. The AHE process can be understood in different ways. One viewpoint starts by creating a histogram of the grey levels (GLs) in a window around each pixel. The input pixel GLs are mapped to the output GLs using the cumulative distribution of GLs, which is the cumulative total over the histogram. If a pixel in a window has a GL value that is lower than all other pixels in the window, the output is 100% black; if it has the window's median value, the output is 50% grey. (Stark, 2000). The primary drawback of the AHE approach is its propensity to excessively amplify noise in relatively homogeneous areas of an image. A novel method known as contrast limited adaptive histogram equalization (CLAHE) is employed to get around this flaw.

Contrast limited Adaptive histogram equalization (CLAHE)

Contrast limited adaptive equalization is a modified part of adaptive histogram equalization. This method applies an enhancement function to all nearby pixels and derives a transformation function. Because of its contrast limiting nature, this differentiates from AHE. (Galgotias College of Engineering and Technology et al., 2014) . This functionality can also be used to create contrast-limited histogram equalization for global histogram equalization (CLHE). In actual reality, this CLHE approach is rarely applied. In the case of CLAHE, the contrast limiting technique, from which a transformation function is produced, must be applied for each neighborhood. This method was established primarily to avoid amplifying noise produced by AHE. (Magudeeswaran & Singh, 2017). On medical imaging, CLAHE yields good outcomes.

2.3.2.4. Image augmentation

In order to artificially increase the size of the training dataset and prevent over-fitting, data augmentation is a typical pre-processing strategy in many machine learning problems, such as image classification. It is an automatic method of increasing the variety of images that will be utilized to train the deep learning algorithms. The dataset must be enormous in order to successfully train a CNN model from scratch. To generate new images, common ways include flipping them horizontally or vertically, rotating them at an angle, scaling them inward or outward, randomly cropping them, translating them, and adding Gaussian noises are used to prevent over-fitting and improve learning ability.

2.3.1 Feature extraction

In the field of computer vision or image processing, features are crucial for identifying relevant data. The most distinctive aspects in a dataset (such as images, texts, and speech) are extracted and used to represent and characterize the data. Finding and extracting features that can be employed to interpret a given image's meaning is the goal of feature extraction. There are numerous techniques for extracting features, depending on geometrical characteristics, statistical features, texture features, and color features. Each primary feature type is further broken into numerous subtypes, for example, color features are further separated into three categories (Color moment, Color histogram and Average RGB) (Mutlag et al., 2020; PADZIL, 2016).

I. Texture Feature Extraction

Texture is a particularly intriguing image feature that has been applied to content-based image retrieval and utilized to characterize images. Repetition of a pattern across a specific area in an image is the primary property of texture. (R. Kumar, 2012). There are numerous ways to extract texture features. Below, we cover three methods for extracting texture features.

Gray Level Co-occurrence Matrix (GLCM)

A histogram to measure gray values that occur at a given offset on an image. Used to extract texture from a broken tissue image. These are the five different texture features that GLCM offers. Entropy, Contrast, Correlation, Energy and Homogeneity.

a. Entropy

Reflects the texture of the image's complexity or unevenness. It is large when numerous GLCM elements have very low values and the image is not texturally constant. A statistical measure of randomness be utilized to distinguish the texture of an input image. Equation 2.1 explain the Entropy.

$$\text{Entropy}= - \sum\sum q\,(i, j) \log q\,(i, j) \qquad (2.2)$$

Where q is the number of gray-level co-occurrence matrices in GLCM

b. Contrast:

This GLCM difference moment statistic quantifies the spatial frequency of a picture. It distinguishes between a group of pixels' highest and lowest values that is continuous. It determines how many local variations are there in the picture. Low spatial frequency and a low contrast picture exhibit the GLCM concentration term around the major diagonal. Calculates the density contrast throughout the entire image, including adjacent and neighboring pixels. The contrast is explained in Equation 2.2.

$$\text{Contrast}= \sum (i, j)^2\, q\,(i, j) \qquad (2.3)$$

Where, q (i, j) = pixel at location (i, j).

c. Correlation

The resemblance of elements of a co-occurrence matrix element to a row or column is measured using this metric. If the gray levels of the pixel pairings are highly correlated, then correlation should be high. This scale's purpose is to quantify the likelihood that the provided pixel pairings would appear as described in Equation 2.3.

$$\text{Correlation} = \frac{\Sigma_{i=0}^{M-1}\ \Sigma_{J=0}^{M-1}\ (i-n_i)\ (j-n_j)\ q\ (i,j)}{\sigma i \sigma j} \qquad (2.4)$$

d. Energy:

This is the summation of squared elements in the GLCM. It is sometimes referred to as the uniformity or angular second moment.

$$Energy = \sum\sum \text{q (i, j)}^2 \qquad (2.5)$$

e. Homogeneity

It used to measure the approximation of the distribution of elements in the GLCM to the GLCM diagonal, which define in Equation 2.5.

$$\text{Homogeneity}=\sum_{i,j} \frac{q(i_1 j)}{1+|j-i|} \qquad (2.6)$$

II. Color Features Extraction

Due to its reliability, efficiency, and computational simplicity, color is one of the most often employed features in image retrieval. A color model is used to represent the image's color. RGB (red, green, and blue), HSV (hue, saturation, and value), and Y, Cb, Cr (luminance and chrominance) are the most widely used color models. Color histogram is one technique. The distribution of colors inside an image or within a selected area of an image is provided by the color histogram. The important property of histogram is that it is invariant to rotation, translation & scaling.(R. Kumar, 2012)

2.3.2 Image Classification

According to their similarities, photos are categorized into different groups through the classification process then the use of classification algorithms in image processing methods is important. It is employed to categorize the features that are taken from the image into multiple classes according to different characteristics. There are numerous ways to categorize an image. Supervised and unsupervised classification are the two primary approaches for classifying images. Each classification has advantages and disadvantages of its own. With a noisy and fuzzy image, it is challenging to get a better outcome than with a typical image. (Jain & Tomar, 2013). Several classification methods, including Naive-Bayes, Support Vector Machine, k-Nearest Neighbor, Shadow, Minimum Mean Distance (MMD), Neural Networks, Decision Trees, Hidden Markov Model, K-means clustering, and Machine Learning, are available for use in real-world applications. These techniques are suited for various applications, and the classifier is chosen based on how well it performs in a particular application domain.

Supervised learning-based classification: Classification that is based on supervised learning uses data that is produced as a result of domain expertise. Pre-labelled samples are necessary in supervised classification in order to assess an image's accurate categorization. This technique becomes time-consuming because it demands for training or specialized knowledge. Due to this, this strategy may not be appropriate in specific situations. Knowing the spectral characteristics or features with respect to the population of each class is important to come up with a decision rule for categorization. Supervised machine learning methods like linear classifiers, k-nearest neighbor, random forest, decision trees and support vector machines (SVM), are the most often employed standard classification algorithms. (Jain & Tomar, 2013)

Unsupervised Classification: Only image attributes are utilized as the several groups of randomly collected data in some situations, and the groups are then mechanically separated into the same classes by utilizing clustering techniques. In these situations, little information about the area needs to be classified. These grouped classifications were subsequently utilized to calculate population statistics. Unsupervised classification is the term used to describe this type of classification. However, since this method does not provide any form of instruction, it demands extensive knowledge of the field or of the approach that is most suited for the target field. And with large data sets computation time is large and it creates useless classifier. (Khanum et al., 2015).

2.3.2.1 Machine learning techniques

Support Vector Machine

The Support Vector Machine (SVM) method is one of the most broadly used machine learning techniques. It is a supervised learning technique used for outlier detection, regression, and classification. SVM concept is straightforward, SVM's main principle is to select the hyperplane that lies "exactly in the middle" of the two classes. The algorithm creates a line or a hyperplane which separates the data into classes. At first approximation what SVMs do is to find a separating line (or hyperplane) between data of two classes. Mathematically speaking, SVM picks the hyperplane that maximizes the shortest distance between hyperplane and all of the classes. In other words, the hyperplane is equidistant from the training data point of the two classes that are closest to it. In SVM terminology, two times the distance between the hyperplane and the nearest point to it is defined as the margin. SVM is hence also referred to as a maximum margin classifier. It is crucial to maximize the margin or, alternatively, choose that specific hyperplane between the training data points of the two distinct classes. (Khanum et al., 2015). SVM offer benefits like effective in high-dimensional spaces and even in situations where number of dimensions is greater than the number of samples. However, if the number of features is much greater than the number of samples, avoid over-fitting in choosing kernel functions and regularization term is crucial.

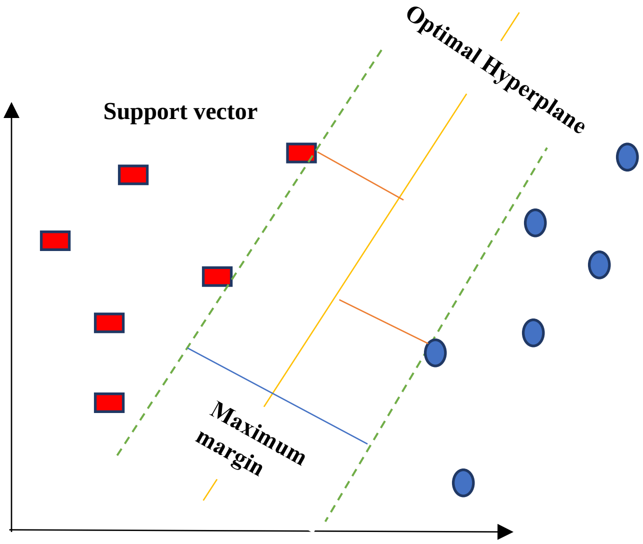


Figure 2.1: SVM algorithm with hyper-plane separating data points from two classes

K- Nearest Neighbor Classification

The K-Nearest Neighbor (KNN) Algorithm, based on the supervised learning approach, is the most straightforward of all machine learning algorithms used for regression and classification. (Jadhav & Channe, 2016). It is based on the principle that the samples that are similar, generally lies in close vicinity. A data sample X is classified by first locating its K nearest neighbors, after which X is given the class label that the majority of its neighbors belong to. The performance of the k-nearest neighbor algorithm is also influenced by the choice of k. Because of the present of some noise in the training dataset, a KNN classifier may be vulnerable to over fitting if k is chosen too low. On the other hand, if k is too big, the nearest-neighbor classifier could incorrectly categorize the test sample because some of the data points on its list of nearest neighbors might be located far from its neighborhood. KNN is an instance-based approach of learning. Instance-based classifiers, also known as lazy learners, store all of the training samples but do not create a classifier until a new, unlabeled sample has to be categorized. Compared to eager learning algorithms (such as decision trees, neural networks, and bayes networks), lazy learning algorithms need less computational time during the training phase but more computation time during the classification phase.

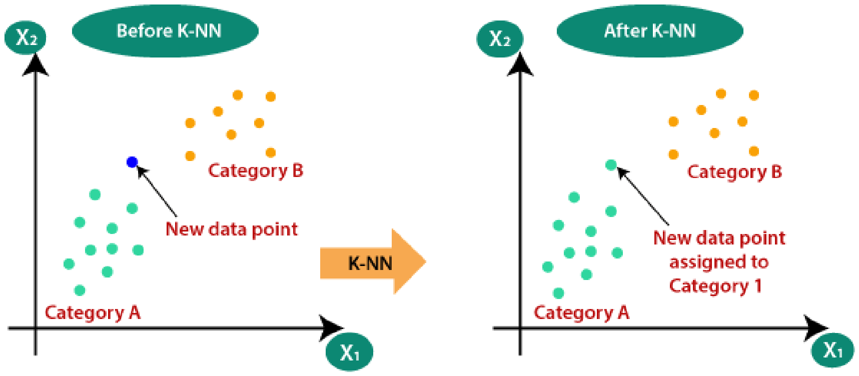


Figure 2.2: K-Nearest Neighbor algorithm(Jaiswal, 2015)

2.3.4.2. Neural network classification

Artificial Neural Network

Statistical learning algorithms called artificial neural networks (ANNs) are motivated by features of biological neural networks. They are employed in a wide range of tasks, from relatively simple classification problems to speech recognition and computer vision. They are implemented as a network of linked processing units, referred to as nodes, that act similarly to biological neurons.

Each node in the network accepts numerous inputs from other nodes and produces a single output depending on the inputs and the connection weights. This output is often sent into another neuron, repeating the process. The input layer, output layer, and hidden layer are the three layers that make up an ANN. The input layer takes the inputs, while the output layer generates an output whereas the layers between these two are referred to as hidden layers. (Sootla, 2015).

Input layer: The input is received by the input layers, which normally have the same number of neurons as the input feature to the network. The output from the input layer is subsequently fed into the hidden layer.

Hidden layer: Between input and output are layers known as the hidden layers. There will be one or more hidden layers in ANN. The learned information acquired from the raw training data is encoded by neural networks using the weight value on the connections between the layers.

Output layer: Depending on the complexity of the task, the output layer may have one or more neurons. Based on the input from the input layer, provide as an output. The neural network's configuration will determine whether the final output is a regression with real values or a collection of possibilities.

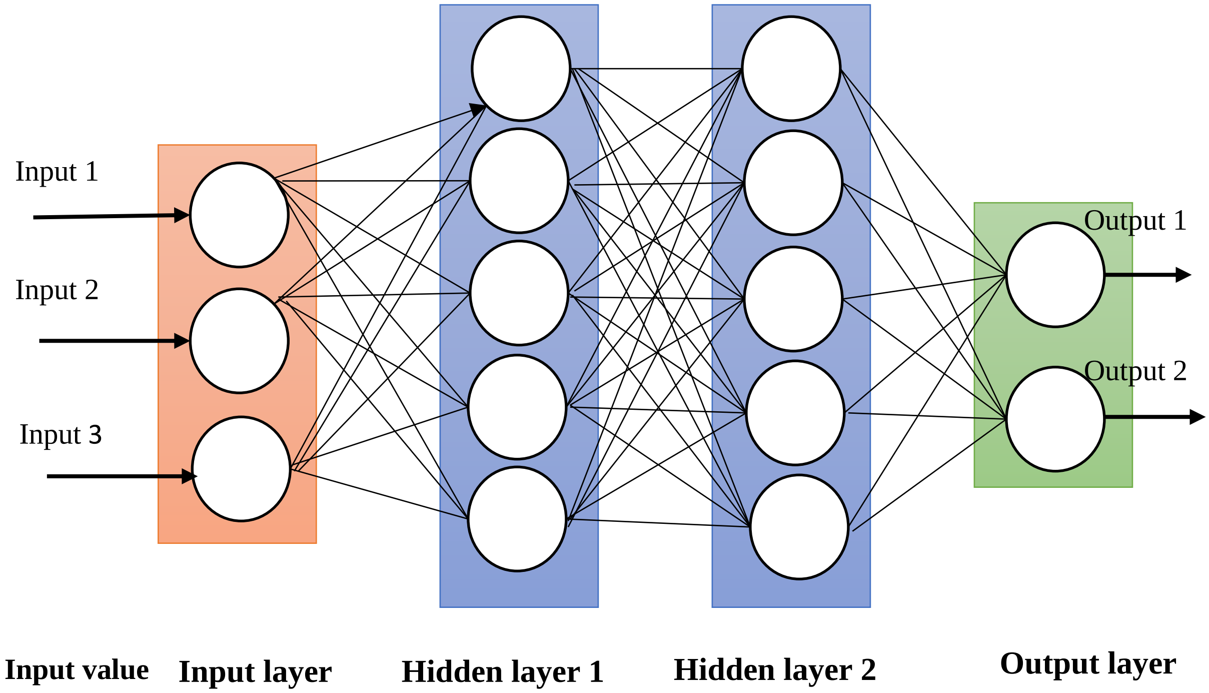


Figure 2.3: The visualization of artificial neural networks architecture

Convolutional Neural Network

Convolutional Neural Network (CNN), also known as ConvNet, is a form of artificial neural network (ANN) that has a deep feed-forward architecture that can learn highly abstracted aspects of objects, particularly spatial data, and more effectively identify them. The visual perception of living things serves as an inspiration for CNN's architecture.

A conventional neural network is composed of one or more blocks of convolution and pooling layers followed by one or more fully connected (FC) layers, and an output layer. A combination of the feature extraction and classification components makes up CNN. Convolutional and pooling layers both extract features. From lower to higher level convolution layers, numerous features are detected. The FC layers are added as a classifier on top of these characteristics, and a probability is generated for the input image. In addition to layer design, a number of other factors, such as activation function, normalization method, loss function, regularization, optimization, and processing speed, among others, affect how well CNN performs. (Ghosh et al., 2019).

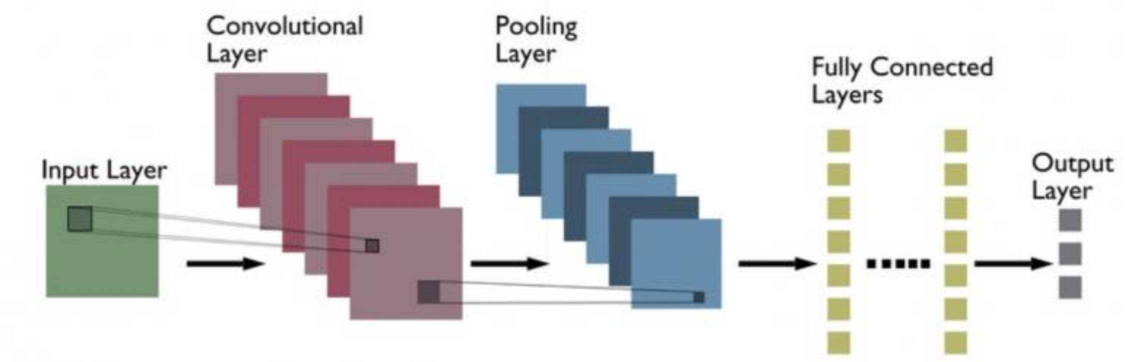


Figure 2.4: basic CNN architecture(A. Kumar, 2022)

a) Convolutional Layer

The most crucial and fundamental element of any CNN design is the convolutional layer. Learning feature representations of the input is the goal of this layer. The learnable convolution kernels or filters utilized in the convolutional layer are used to construct various feature maps. A kernel can be thought of as a grid of discrete values or numbers, each of which represents the kernel's weight. Each feature map unit is linked to a preceding layer receptive field. The input and kernels are convolved to create the new feature map, which is then processed with applying an elementwise non-linear activation function. After that, the weights are adjusted and the kernel learnt to extract useful features with each training period(epoch).

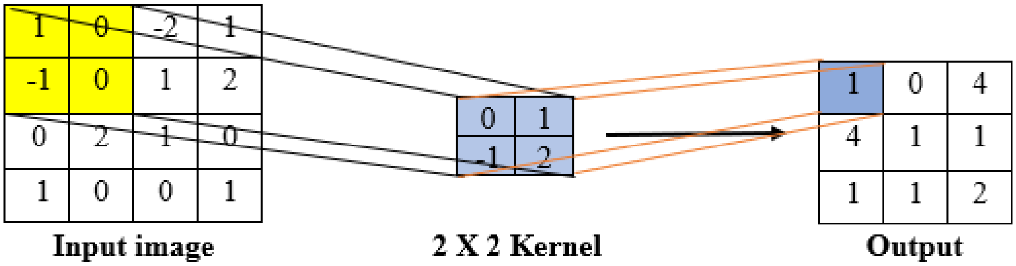


Figure 2.5: kernel size of 2 x 2 Convolution operations

b) Activation Layer (Non-Linearity)

Even though they are called activation layer they aren't considered to be "layers" in the traditional sense, since no parameters or weights are learned within an activation layer. The activation function in a CNN's non-linearity layer takes the feature map produced by the convolutional layer and produces the activation map as its output, where the input value is derived by computing the weighted sum of neurons' input and then adding bias to it (if there is a bias). In other words, by generating the corresponding output, the activation function determines whether or not a neuron will fire for a given input. In the CNN design, non-linear activation layers are utilized after each learnable layer (weighted layers, such as convolutional and FC layers). Non-linear operations can take a variety of forms, but the most well-known ones include Sigmoid, Tanh, and Rectified Linear Unit (ReLU). The input for the sigmoid activation function is a real number, and the output is bound to the range [0, 1]. However, a very unfavorable characteristic of sigmoid is that the gradient becomes almost zero when the activation is at either tail. A real-valued number is condensed by Tanh to the range [-1, 1]. Similar to sigmoid neurons, the activation saturates, but its output is zero-centered. The ReLU activation function is the one that Convolutional Neural Networks employ the most frequently. All input values are changed to positive ones using this technique. ReLU speeds up convergence by six times and is more dependable than sigmoid and tanh. ReLU has the benefit of requiring a much lower computational load than other methods. ReLU can, unfortunately, be brittle during training, which is a drawback. It can be updated by a strong gradient that prevents the neuron from ever updating further. However, by choosing an appropriate learning rate, we can make this work.(Mayank Mishra, 2020).

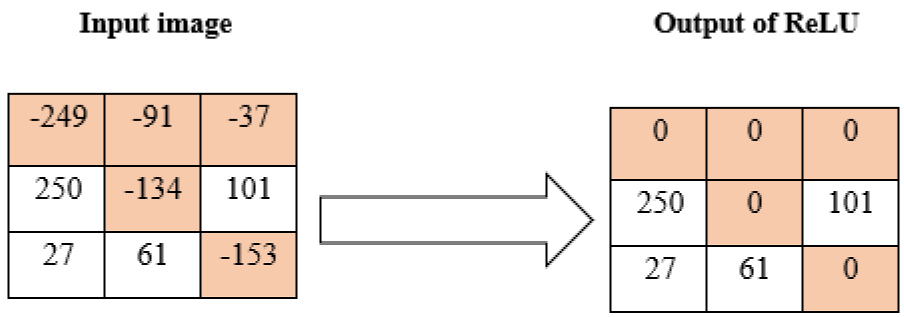


Figure 2.6: The operation of ReLU activation function

c) Pooling Layer

The feature maps (generated after convolution operations) are sub-sampled using the pooling or sub-sampling layers. These layers either take a small section of the convolutional output as input and down sample it to produce a single output. The biggest disadvantage of the pooling layer is that occasionally it causes CNN's overall performance to suffer. The reason for this is that the pooling layer enables CNN to determine if a particular feature is present in the input image or not without worrying about the feature's exact location. Different pooling approaches, such as max pooling, min pooling, average pooling, gated pooling, tree pooling, etc., are employed in various pooling layers. The most popular and frequently employed pooling method is Max Pooling. While average pooling takes the average of the specified pixels, max pooling chooses the highest value of the specified pixels. (Ghosh et al., 2019).

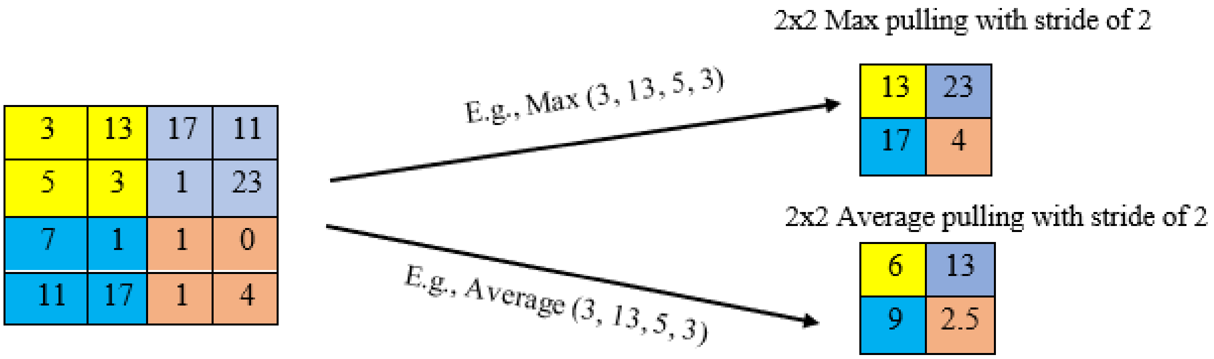

2x2 Max pulling with stride of 2
3
13
17
11
5
3
1
23
7
1
1
0
11
17
1
4
E.g., Max (3, 13, 5, 3)
13
23
17
4
2x2 Average pulling with stride of 2
E.g., Average (3, 13, 5, 3)
6
13
9
2.5

Figure 2.7: Effect of max pulling and average pooling operation

d) Fully connected Layer

Fully Connected Layer is simply, feed forward neural networks. Every CNN architecture's final part or layer, which is used for classification, typically consists of fully-connected layers, where each neuron in a layer is linked to every other neuron in the layer above it. The last layer of Fully-Connected layers is used as the output layer (classifier)of the CNN architecture. The final output of CNN is produced by the FC layer, which receives input from the final convolutional or pooling layer, which is composed of a set of metrics (feature maps). These metrics are flattened to create a vector, which is then fed into the FC layer.(Ghosh et al., 2019)

Dropout Regularization

Over-fitting is the primary obstacle to a CNN model achieving adequate generalization. A model is said to be over-fitted if it performs remarkably well on training data but poorly on test data (unseen data). Regularization employs a number of intuitive concepts to assist prevent over-fitting. Dropout is a regularization method that ignores randomly chosen neurons during training in neural networks to reduce overfitting. They "drop-out" at random. This means that any weight updates are not applied to the neuron on the backward pass and that their effect to the activation of downstream neurons is temporally removed on the forward pass.

Optimizer

Any supervised learning algorithm's primary goal is to reduce Error (difference between the predicted output and the actual output), also known as the loss functions, which is based on a variety of learnable factors like weights, biases, and other variables. The gradient-based learning approaches are an obvious choice for learning to a CNN model. The model iteratively searches for the locally optimal solution in each training session while continuously updating the model parameters to lower the error. The duration of parameter updating steps is referred to as the "learning rate," and a whole parameter update iteration that uses the entire training dataset once is referred to as a "training epoch". Although the learning rate is a hyper-parameter, it must be carefully chosen so as to not negatively impact the learning process. Gradient descent is used by the majority of neural network-based methods, including CNN, to reduce error rates during training and to restructure internal parameters. First-order optimization algorithm gradient descent gives direction and an increasing or decreasing error function through its derivatives. The error function is guided by information and is modified to the local minimum.

Optimizers are algorithms or techniques that alter the weights and learning rates of our neural network in order to lessen losses. The optimizers we employ determine how we should modify the weights or learning rates of our neural network to minimize losses. Reducing losses and delivering the most accurate outcomes possible are the goals of optimization algorithms or strategies. (Yaqub et al., 2020). Popular optimizers of CNN state-of-the-art gradient descent-based optimizers, namely Adaptive Gradient (Adagrad), Root Mean Square Propagation (RMSProp), Adaptive Momentum (Adam), Adaptive Max Pooling (Adamax) and Nesterov-accelerated Adaptive Moment Estimation (Nadam) are discussed below.

Adam

The Adam technique calculates the adaptive learning rate for each gradient training parameter. It is a relatively straightforward method using first-order gradients that is computationally efficient and has a low memory demand for stochastic optimization. This method is used to calculate learning rates independently for different parameters from approximations that include first- and second-order moments in the case of machine learning problems with high-dimensional parameter spaces and large data sets. Adam lowers the computational cost, uses less memory during implementation, and is insensitive to the gradients' diagonal rescaling.

Adamax

Adam is the model for Adamax, with modifications made to the way the infinity norm is applied.

Adagrad

Adagrad modifies the learning rate in accordance with the parameters, making larger updates for parameters that are inconsistent and fewer updates for values that are sequential. Adagrad eliminates the need for a manual learning rate adjustment, but it also slows intermingling speed and causes the learning rate to shrink as a result of its collection of squared gradients in the denominator.

RMSProp

By employing a moving average of the squared gradient, which normalizes the gradient by taking into account the size of recent gradient descents, RMSProp attempts to address Adagrad's drastically decreasing learning rates. As a result, the algorithm would progress in a horizontal direction with larger steps converging faster as the learning rate increased.

Nadam

An improvement in the performance of the optimization method can be obtained by using Nadam, an extension of the Adam algorithm that integrates Nesterov momentum. Much like Adam is essentially RMSprop with momentum, Nadam is Adam with Nesterov momentum.

Today, there are many other types of CNN models, like as LeNet, AlexNet, ResNet, GoogleNet / Inception, Visual Geometry Group (VGG), DenseNet, etc., that have been effectively constructed and applied in a variety of image processing and object recognition fields.

2.4. Related work

(Elangovan & Nath, 2021) design glaucoma assessment from color fundus images using convolutional neural network. For automatic categorization of glaucomatous and normal fundus images, they have developed an 18-layer CNN model with four convolutional layers, two max pooling layers, and one fully connected layer. In the pre-processing stage, only data augmentation and image resizing were used. However, they might improve their model and boost the model's accuracy by using a better preprocessing technique, such as improved noise reduction and contrast enhancement techniques. They employed CNN to extract the important features from the fundus images, and they used CNN’s parts like convolution, batch normalization, ReLU, and max pooling to extract the features. Fully connected and softmax layers were employed to distinguish between glaucomatous and normal images. To test the model's effectiveness, they trained it using a number of databases, including DRISHTI-GS1, ORIGA, RIM-ONE2, ACRIMA, and large-scale attention-based glaucoma (LAG). And they achieved overall accuracy for the DRISTHI-GS1, ORIGA, RIMONE2, LAG, and ACRIMA databases of 86.62 percent, 78.32 percent, 85.97 percent, 94.43 percent, and 96.64 percent, respectively.

(Raghavendra et al., 2018) developed deep convolution neural network for accurate diagnosis of glaucoma using digital fundus images. They presented a CNN model with eighteen layers. They use a dataset prepared by themselves with a total of 589 normal images and 837 glaucoma images. Only resizing of the dataset dimension in to 64x64 was performed in the preprocessing step before feed in to their CNN model. Data preprocessing is essential before its actual use because data must be in a format appropriate for ML. The dataset is preprocessed in order to check missing values, noisy data, and other inconsistencies before executing it to the algorithm. Additionally, CNN needs the maximum number of images to perform at its best. However, because they only used a small number of datasets for the specified input image dimensionality, it is not possible to generalize the system with a smaller number of images. Because in order to evaluate a deep learning model, CNN model need more image dataset in order to learn best features of the given image. When the number of image input samples are more, taking a smaller sized image as an input is affordable as the network will train with greater number of images. Their model attained the best performance for 0.001 learning rate and their method obtained accuracy, sensitivity, and specificity of 98.13%, 98%, and 98.30%, respectively, for the maximum number of the given images.

(Saxena et al., 2020) presented a six-layer CNN model with four convolutional layers and two layers of fully connected for the detection of glaucoma. Before feed in to the model they performed two preprocessing steps. The first one was extraction of Region of Interest (ROI) of the fundus image using ARGALI: (An Automatic Cup-to-Disc Ratio Measurement System for Glaucoma Analysis Using Level-set Image Processing). In ARGALI method, the preprocessing was used for the removal of the bright fringe, which will help in getting the center for the trimming circle and the radius of the trim. Then the obtained ROI was fixed to 256*256 resolution. In the end, for all the pixels of the disc image, the mean value was subtracted from every pixel for the removal of the illumination in the pictures. The other one they used in the preprocessing step to make the model not to suffer from overfitting was data Augmentation. But other preprocessing techniques which have high impact on the performance of generalization didn’t consider in this work. Two datasets were used in this study; like ORIGA and the SCES dataset and They obtained (Area Under Curve) AUC for the proposed method 0.0822 for the ORIGA dataset, and for the SCES is 0.882.

(Sallam et al., 2021) early detection of glaucoma using transfer learning from pre-trained CNN models proposed in this study. Pre-trained AlexNet, VGG11, VGG16, VGG19, GoogleNet (Inception V1), ResNET-18, ResNET- 50, ResNET-101 and ResNet-152 models were leveraged to develop the proposed study. For this research they had used LAG dataset and to detect Glaucoma by the proposed transfer learning models, only data augmentation on the training data was first applied. In this study total number of 2631 images for training, 751 images for validation, and 396 images for testing were used and trained on a learning rate of 0.001 by taking 32 samples for training, 32 samples for testing and 32 samples for validation in each epoch. Then after transfer learning was trained through forward and backward propagation. Results of 81.4%, 80%, 82.2%, 80.9%, 82.9%, 86.7%, 85.6%, 86.2%, and 86.9% were observed on LAG dataset using AlexNet, VGG11, VGG16, VGG19, GoogleNet (Inception V1), ResNET-18, ResNET-50, ResNET-101 and ResNet-152 models respectively. As their experimental results indicated, out of these results, the ResNet-152 model achieved a high accuracy with precision of 86.9% and recall of 86.9%, which is the best compared to the other models. But they could enhance those models' performance by using different preprocessing techniques like noise removal, contrast enhancement, and so on, and also, they used pre-trained model for classification.

(**Afroze et al., 2021**) develop CNN based Inception V3 model. For this research, they used a total of 6072 images by combining collected images from the ACRIMA dataset, the LAG dataset and the Glaucoma dataset. Of those, 3736 were normal fundus images and 2336 were glaucomatous. The dataset was then split into training and test sets of images then the Inception V3 model, a pre-trained model, was utilized to extract features from the datasets in order to perform the study. The model was then trained after applying rotation, width shift, height shift, zooming, sheer, channel shift, and horizontal shift augmentation techniques using the Image Data Generator class of Keras. After that, for the classification, they used a CNN model which had a total of 312 layers: 1 input layer, 94 Cov2d layers, 94 batch normalization layers, 94 activation layers, 11 mixed layers, 8 average pooling layers, 4 max pooling layers, 2 concatenate layers, 3 global average pooling layers, and 1 dense layer. During the training time, 600 images were taken for the purpose of validating the model from training images. Then they evaluated it using test images. They also evaluated the impact of other different pretrained models like ResNet50 and DenseNet121 other than Inception-V3. As their experimental results indicated, Inception-V3, along with the other listed pretrained models, gave the best result in test accuracy of 0.8529. The gap in this study is that it is not appropriate to combine different datasets and use them as one large dataset. It is because different datasets have their own properties, from the camera they take up to the knowledge and experience of the expert ophthalmologist who labeled the dataset. The other limitation of this study is that they don't use a preprocessing technique to enhance the model and they use a pre-trained feature extractor.

(Vaghjiani et al., 2020) proposed a method for visualizing and understanding inherent image features in CNN-based glaucoma detection. They conducted this study to determine the relative importance of the optic disc compared to its surrounding region for glaucoma detection in addition to the detection of glaucoma using VGG16. A VGG16 has been trained to perform glaucoma detection using cropped fundus images. In their experiment, stochastic gradient descent with a learning rate of 0.0001 and momentum of 0.9 has been used as the optimizer, and binary cross-entropy has been used as the loss function during training. They used a total of 1817 fundus images from six publicly available datasets: ACRIMA, Drishti- GS, HRF, RIM-ONE r2, sjchoi86 HRF, and DRIONS-DB. The data set was divided into two parts; 80% of the images were used for training and the remaining 20% were used for testing. They had obtained an accuracy of 93% with a sensitivity and specificity of 92% and 94%, respectively. In other experiments, in order to investigate the relative importance of the OD and surrounding region for glaucoma detection, they trained and validated the VGG16 model independently under two different scenarios. In the first case, cropped images have been used without introducing any image preprocessing. In the second case, the images were preprocessed to mask out the regions outside the optic disc. For case-I they had obtained an accuracy of 92.7% and for case-II they had obtained an accuracy of 91.2%. The area under the ROC curve (AUC) scores for case-I and case-II were 97.4% and 96.5%, respectively. From those results, they confirmed the hypothesis that most of the important information for detecting glaucoma is contained inside the optic disc. Despite the fact that they examined a good experiment, it is difficult to decide the diameter for the cropping region of interest as the size of the optic disk depends on the image taking angle, and it is better to apply different preprocessing techniques to improve the postprocess.

2.4. Summary

In this chapter, we reviewed different studies which are related to our study on detection of glaucoma using image-processing and deep learning techniques. We tried to show and explain the performance and the limitations of each study and how we are going to fill these gaps. Most researches that were reviewed in this paper to identify glaucoma from fundus image using CNN is without using a better pre-processing method to enhance the images and increase the detection accuracy. In the researches there are gaps like use small number of datasets, most researchers we review uses a pre trained CNN model and most of them don’t use good preprocessing methods.

Preprocessing techniques are not considered which have high impact on the classifier like resizing with good method, better noise removal techniques and also better contrast enhancement.

In our research we have used image preprocessing, data augmentation and features extraction and classification using different machine learning approaches including self-build CNN model in order to compare and evaluate the impact of the model on the given dataset. Image resizing was the first step that was done in this study using bicubic interpolation which then followed by removal of noise from the image using median filter and for contrast enhancement we use CLAHE in order to get an enhanced image for better classification. Data augmentation also takes place in order to balance the datasets and also to reduce overfitting of the model. What makes our research different from those stated above is we have design and developed our own CNN model to detect glaucoma with better preprocessing methods that are listed above and get a better accuracy even than a pre trained model that used in (Sallam et al., 2021) and (Vaghjiani et al., 2020).

CHAPTER THREE

SYSTEM DESIGN AND MODELING

3.1. Introduction

In this chapter, a detailed description of the proposed system or model for the detection of glaucoma is discussed. We proposed an architecture that will process the digital fundus image. It requires to pass via a series of steps starting from digital image acquisition to image preprocessing (including image resizing, noise removal, contrast enhancement, and image augmentation), Feature extraction, and classification. The proposed model has been trained by a publicly available digital optic fundus image dataset. Later on, feature extraction for the fundus image was performed using CNN, CNN+GLCM, and GLCM to select the best one. Finally, classification was done using tree different classification algorithms (SVM, CNN and KNN) to compare the best classification performer algorithm. Discussion of different evaluation metrics to test the developed model are the other topic we covered in this chapter. A general description of the proposed system architecture is presented in section 3.3.

3.2. Fundus image Data Set

Enough picture capture is the very first and crucial step because many classification methods require a huge amount of labeled data to achieve a better outcome. Finding and gathering a better digital optics fundus image dataset was the first and most crucial thing we did to finish this thesis (which is labeled and organized by Ophthalmologists). A fundus camera or retinal camera is a specialized low power microscope with an attached camera designed to photograph the interior surface of the eye, including the retina, retinal vasculature, optic disc, macula, and posterior pole (i.e., the fundus). (Hani et al., 2013). We have trained our model using the large-scale attention-based glaucoma (LAG) fundus image database which are publicly available (which is based on request) fundus images respiratory.

The database's creator (L. Li et al., 2019) claims that 10,861 people are participating in the LAG database, of whom 10,147 people are collected with just one fundus image or one image per eye and subject. The remaining people make reference to multiple pictures of each subject. The Chinese Glaucoma Study Alliance (CGSA) and Beijing Tongren Hospital provided the fundus images for the LAG database. The total number of the dataset is 11,750 with this number 4878 images belonging to positive glaucoma and 6882 belonging to negative glaucoma. However, the one that we found and which is freely available up on request was LAG database part one which contains a total of 4814 fundus images (i.e., 1711 glaucomatous images and 3143 healthy images). The average age of this LAG database is 53.6, with a standard deviation (SD) of 15.3, and there are 6,300 female patients (or 53.6%). They use three different kinds of cameras for their fundus images: Topcon, Canon, and Carl Zeiss. Additionally, the input fundus image dimensions range from 582 x 597 to 3456 x 5184, with an average of 1977 x 2594 and a standard deviation of 840 x 1417. Those images are in jpg format and need to resize so we use bicubic resizing technique, to make all images have the same size which enhances the processing time and accuracy of the model. The reason why we resize those images into the same size and aspect ratio is that the neural network models assume a square shape input image and raw data have a large size that is difficult to process.

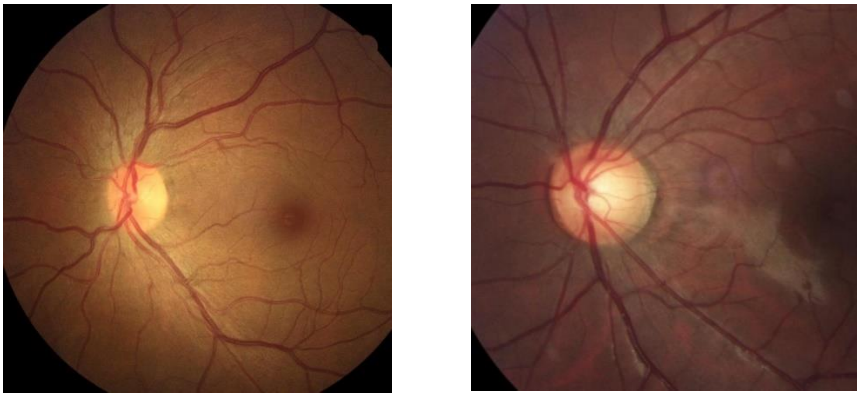


Figure 3.1: (a) Healthy fundus image (b) Glaucomatous image

3.3. System Architecture

The proposed system has three components: preprocessing, feature extraction and classification as shown figure 3.2

3.4. Image Preprocessing

The dataset must first be processed by image processing before moving on to the next stage. At this level, several image processing techniques are used to improve the accessibility of the provided dataset for our model. In the present study, image resizing, noise reduction, contrast enhancement, and augmentation were the image pre-processing techniques applied. The following subsections discuss the preprocessing algorithms employed in this study. In the fundus image the

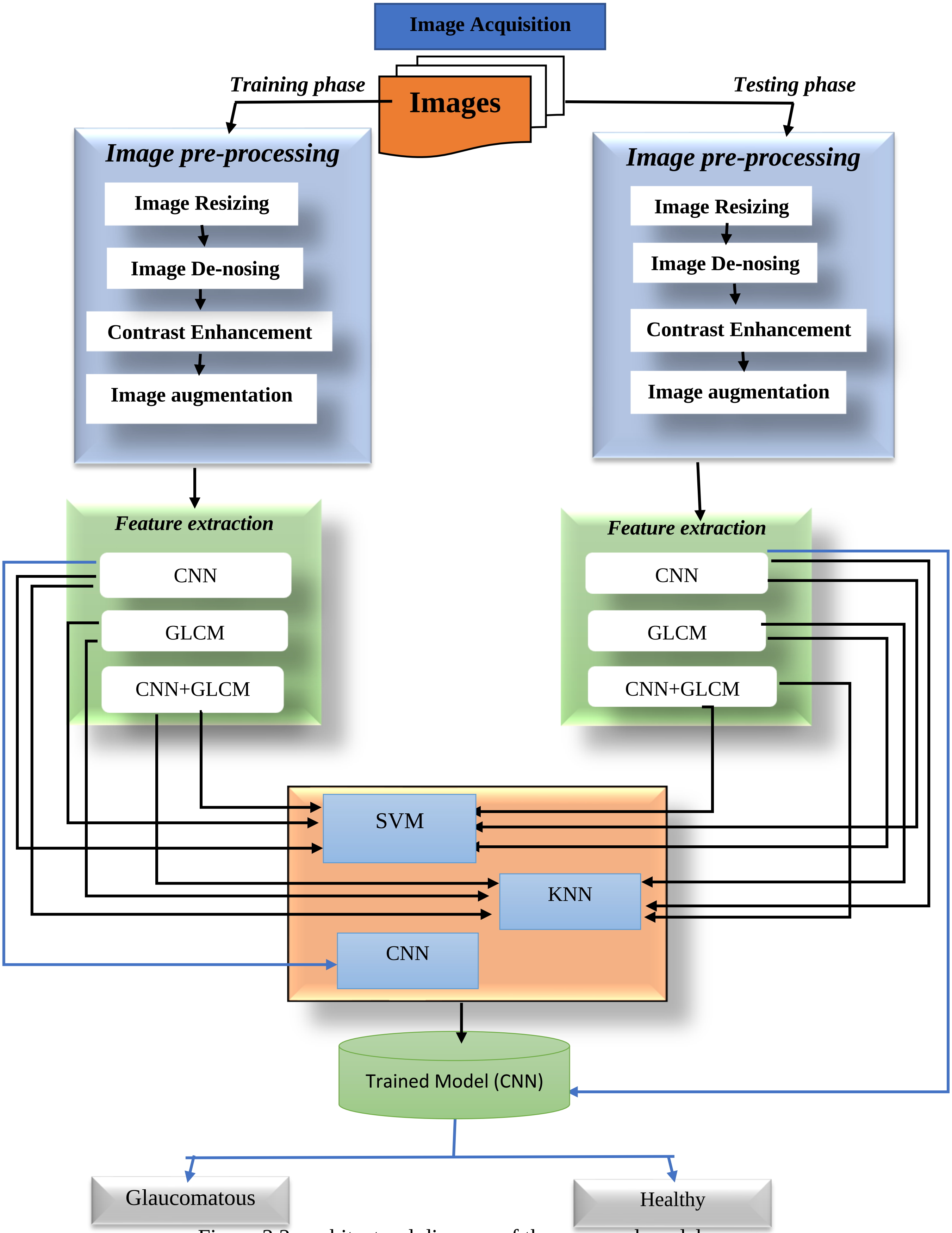


Figure 3.2: architectural diagram of the proposed model

glaucomatous image is differentiated from that of the healthy is by the bright area inside the optic cup. So, the brightness of the fundus image plays an important role in the detection of the disease and then needs a better image processing method to make the fundus image meaningful to the classifier that’s why we use histogram equalization.

In the fundus image the glaucomatous image is differentiated from that of the healthy is by the bright area inside the optic cup. So, the brightness of the fundus image plays an important role in the detection of the disease and then needs a better image processing method to make the fundus image meaningful to the classifier that’s why we use histogram equalization.

3.4.1. Image resizing

In order to reduce the computational time of the model, resizing was applied to all of the provided images to make the image dimensions similar. Since the given image dataset's dimensions vary, as we discussed in section 3.2, it is required to first normalize the images. To some extent, the images were compressed in order to achieve the basic requirements for clarity while speeding up the classification process's subsequent steps. The 96 by 96 image size was chosen for this study after testing and comparing many different sizes in order to reduce computation for both feature extraction and identification. For this study, cubic, bi-cubic, nearest had been considered in order to identify the suitable resizing techniques. As indicated (Chitradevi & Srimanthi, 2014) the size of an image has its own effect on the feature extraction as a result, therefore selecting a resizing technique is very important in image processing. Bi-cubic interpolation is therefore used to resize the input image to 96 x 96 since, according to (Sinha et al., 2014), it is smoother and does not blur the image even after multiple interpolations.

Algorithm 1: Resize

```
Input: An input Image
Output: Resized_Image
begin
        Image = original Image;
        Resized_Image = resize (Image, (96, 96), Image.Bicubic);
        Return Resized_Image;
End
```

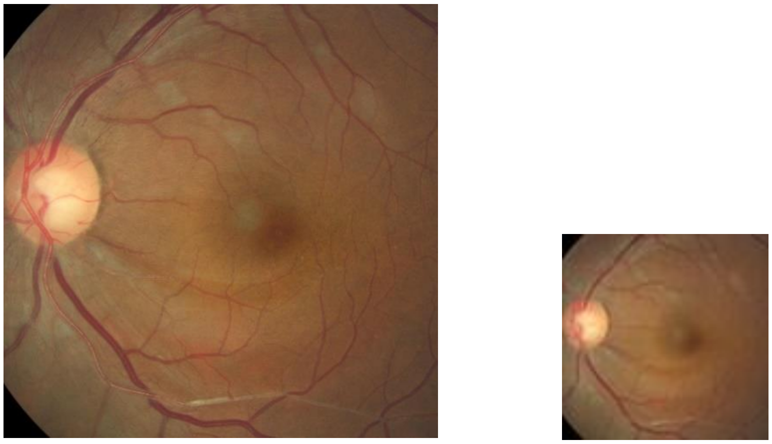


Figure 3.3: (a) original glaucomatous image (b) result of the resized glaucomatous image

Median Filter: - From the filters we tested for this investigation, the median filter was chosen. A non-linear digital filtering method called the Median Filter is frequently used to eliminate noise from an image or signal. From the filters we test, the median filter is chosen for this study is as a result of we get high signal-to-noise ratio (SNR) for our dataset. This kind of noise reduction is a common pre-processing technique to enhance the outcomes of subsequent procedures. The median filter, a straightforward and effective non-linear filter based on order statistics, reduces noise while maintaining relatively sharp edges.

Algorithm 2: Noise removal

```
input: Resized image
Output: Noise removed image
begin
    Receive noisy image;
    Apply 3×3-pixel window median filter;
    Return noise removed image;
End
```

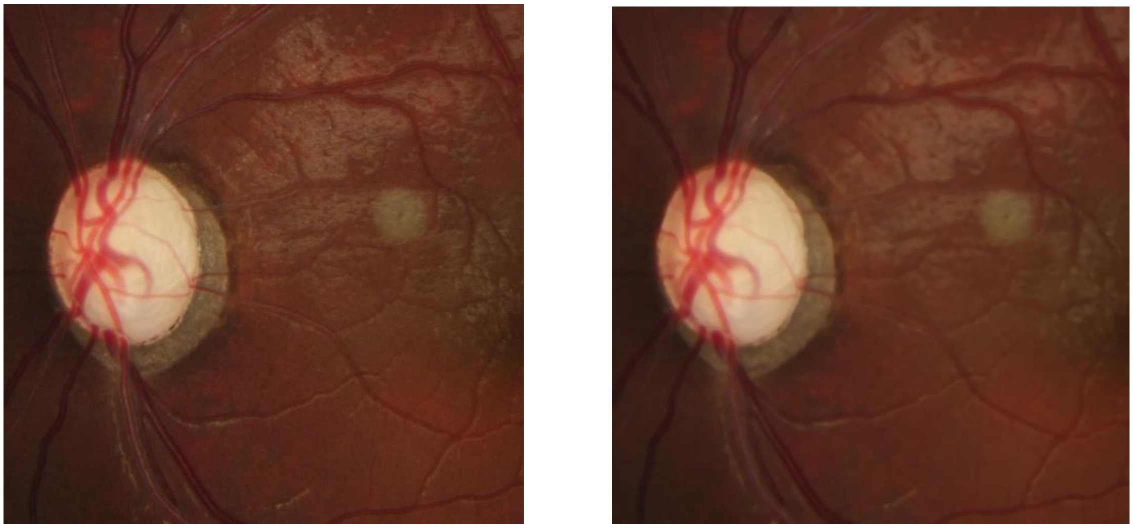


Figure 3.4: (a) original glaucomatous image (b) result of the median filtered fundus image

3.4.2. Contrast enhancement

Image enhancement is employed to accentuate or sharpen the image properties or features of an image such as boundaries, edges, or contrast to make a display more recognizable. Histogram equalization is a nonlinear stretch that redistributes pixel values so that there is approximately the same number of pixels with each value within a range. The main purpose of Histogram equalization the result approximates a flat histogram. Therefore, contrast is increased at the peaks and lessened at the tails. (Chitradevi & Srimanthi, 2014). In the fundus image the glaucomatous image is differentiated from that of the healthy is by the bright area inside the optic cup. so the brightness of the fundus image plays an important role in the detection of the disease and then needs a better image processing method to make the fundus image meaningful to the classifier that’s why we use histogram equalization. HE technique significantly changes the brightness of an image. Hence the output image gets saturated. But there are some limitations are there in HE. To overcome this limitation of HE, in this study, we have used CLAHE instead of basic histogram equalization to obtain better results.

```
Algorithm 3: Histogram equalization
input: Noise removed image
Output: Histogram equalized image
begin
    Receive noise removed image;
    Apply CLAHE;
    Return Histogram equalized image;
End
```

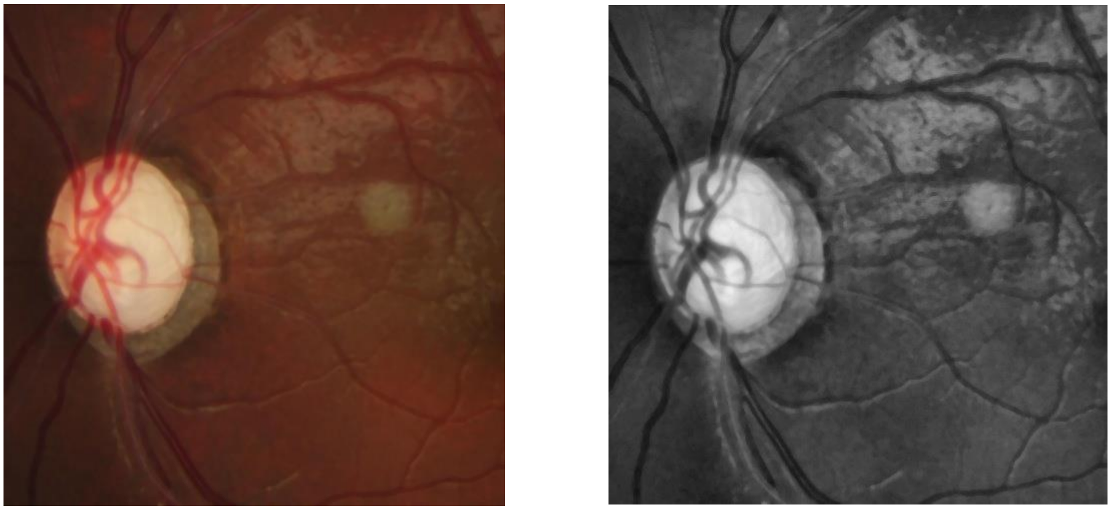


Figure 3.5: (a) filtered glaucomatous image (b) result of the Equalized fundus image

3.4.3. Data augmentation

It is known that training deep neural networks from limited training data causes the over-fitting problem. Data augmentation is a way to reduce over-fitting and increase the amount of training data. It is the process of modifying the training data through random transformations. Data augmentation help to increase our dataset, increasing the size of dataset help to improve model performance. As we mention in the previous section 3.2, the dataset that we found was a total of 4814 fundus images (i.e 3143 glaucomatous images and 1711 healthy images) there is unbalanced number of image dataset in each class. so that we use data augmentation not only to overcome overfitting problem but also to equalize the dataset of the two classes. We have applied transformation such as rotation range (45), width shift range (0.2), height shift range (0.2), shear range (0.2), zoom range(0.2), shear range(0.2) and horizontal flip (True).

Algorithm 4: Data Augmentation

```
Input: Histogram equalized image
Output: Augmented image
begin
    Assign Rotation_Range, Width_Shift_Range, Height_Shift_Range, Shear_Range,
     ZoomRange;
    Set HorizontalFlip (‘true’), FillMode (reflect);
    DataGen = (Rotation_Range, Width_Shift_Range, Height_Shift_Range,
    Shear_Range, Zoom_Range, Horizontal_Flip, Fill_Mode) ;
    Return DataGen;
End
```

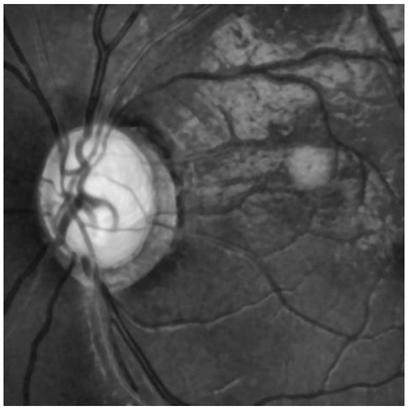
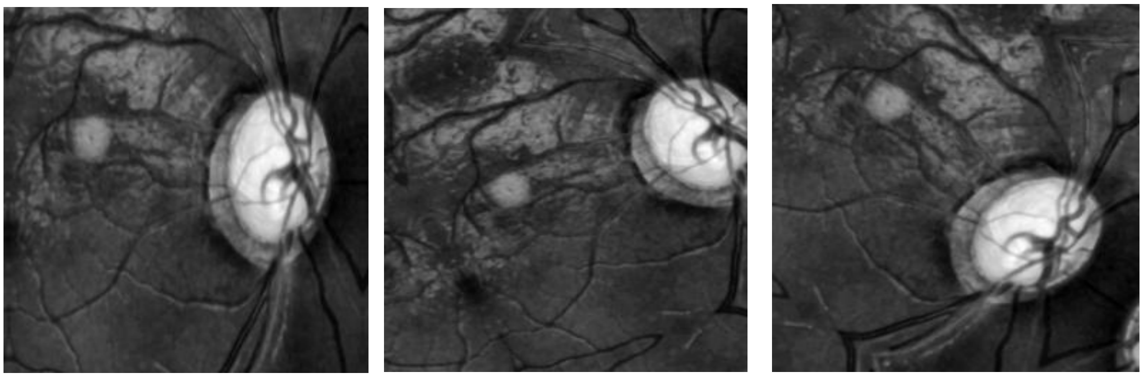


Figure 3.5: original fundus image



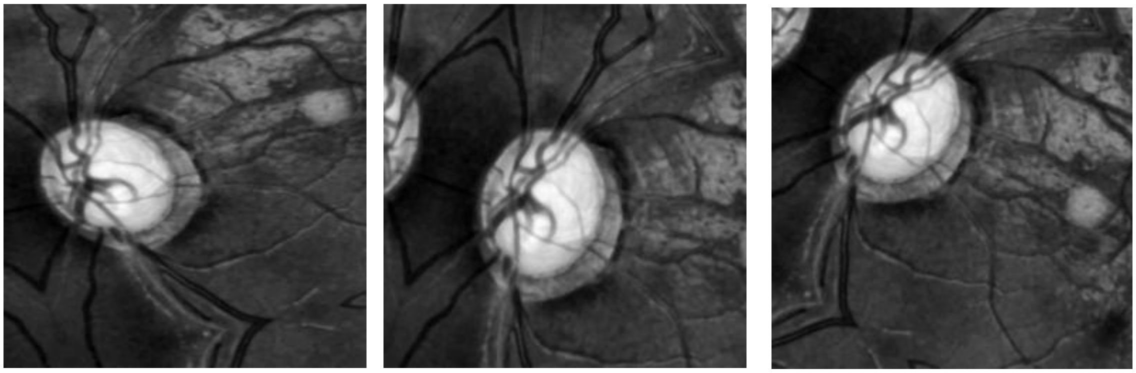
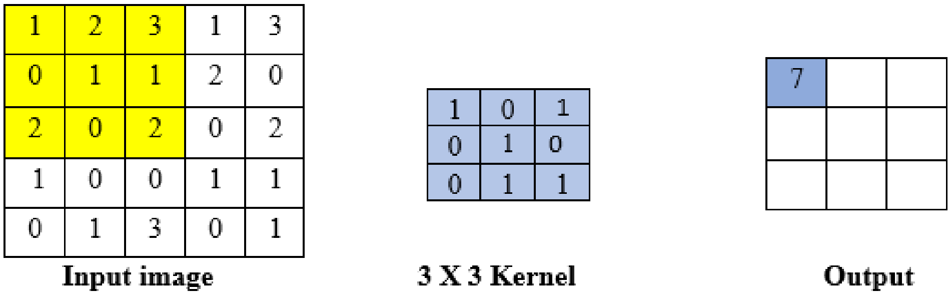


Figure 3.6: Typical results of augmented fundus images

3.5. Feature extraction

A CNN typically includes two primary components: a long chain of convolutional layers for feature extraction and a few (or possibly one) layers of a fully connected neural network for classification. In this study, feature extraction is done using CNN and GLCM to compare the effect on the classification. Our CNN model is constructed using a variety of layers which has six convolutional layers, six ReLU activation layers for achieving non-linearity and max pooling layers for feature reduction. Below, a thorough explanation of the layers employed in our model.

Convolution layer: There are 6 convolutional layers in our model. The convolutional layer's image input is 96 × 96 x 1. Four parameters make up the convolutional operation. The number of filters employed to regulate output volume depth is the first parameter. In this study, we employed filter sizes of 16, 32, 64, 128, 256, and 512. The number of filters we have applied are increased as we go down to the fully connected layers. The kernel size, which controls the size of filters which nearly always square, is the second parameter in the convolutional layer. We have employed kernel sizes of 5 x 5 and 3 × 3. The third parameter, stride size, describes how many pixels are skipped during the convolutional operation. We skipped putting stride size and employ the default stride value (1,1), which causes the filter to shift one pixel to the right for every horizontal movement and one pixel downward for every vertical movement. Padding, the fourth parameter, is used to regulate the output's size. We employ padding that is "same" or zero padding, which means the size of the output is equal to the size of the input if the stride size is one.



1	2	3	1	3
0	1	1	2	0
2	0	2	0	2
1	0	0	1	1
0	1	3	0	1

Input image

1	0	1
0	1	0
0	1	1

3 X 3 Kernel

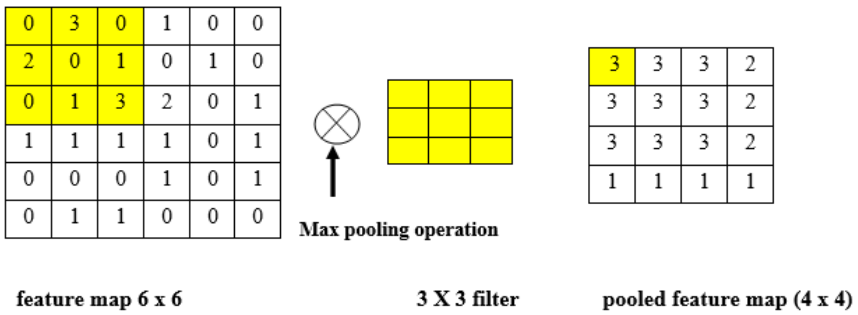
7		

Output

Figure 3.7: Output of convolution with zero-padding and stride size 1 with kernel size 3X3

Activation layer: the output of the activation function is always the same as the size (dimension) of the input. Hence, the width, height, and depth of the output layer is the same as the width, height, and depth of the input layer respectively. In the activation layer of our model, we employed ReLU activation function. The ReLU function's primary benefit over other activation functions is that it does not activate all the neurons at the same time. If the values in the input layer are negative, the ReLU activation function returns zero; otherwise, it returns the value that was previously there. For the negative input values, the result is zero that means the neuron does not get activated. The ReLU function is far more computationally efficient than the sigmoid and tanh functions since it only activates a certain number of neurons.

Pooling layer: The size of the feature maps is reduced by pooling layers. Thus, it reduces the number of parameters to learn and the amount of computation performed in the network. Two common functions that are used in the pooling operation are max pooling and average pooling. Max pooling has been used for our model. The input volume's width and height are gradually decreased using the pooling procedure. Two parameters are needed for the pooling procedure. The first parameter, pool size, determines by how much we want to shrink the input volume's spatial size. Due to the fact that our input image was resized to 96 x 96 pixels, we used a max pooling kernel size of 3 x 3 for our model. The stride size, the second parameter, controls how many pixels are skipped during the pooling operation. Each time we do the pooling operation, we utilize the default stride size of one.



0	3	0	1	0	0
2	0	1	0	1	0
0	1	3	2	0	1
1	1	1	1	0	1
0	0	0	1	0	1
0	1	1	0	0	0

3	3	3	2
3	3	3	2
3	3	3	2
1	1	1	1

Figure 3.8: Input image (left) and output of max pooling (3 x 3) (right)

Algorithm 5: feature extraction

```
Input: Preprocessed image
Output: feature vector
begin
    Get the preprocessed image I;
    Initialize the number of filter K, filter size F, zero padding ZP,
    pooling size PS, Number of node N, dropout D, number of class C;
    for i = 0; i < 2; i++ do
        Apply 3 x 3 convolution operation, convolution1=Convolution (K, ZP, ReLU);
        Apply max pooling operation, max pooling (PS, ZP);
        Apply 3 x 3 convolution operation, convolution2= Convolution (K, ZP, ReLU);
        Apply max pooling operation, max pooling (PS, ZP);
        Apply 5 x 5 convolution operation, convolution3= Convolution (K, ZP ,ReLU);
        Apply max pooling operation, max pooling (PS, PST);
    end
    Apply fully connected layer, fully connected layer (N);
    Apply dropout operation, Dropout (D);
    Apply fully connected layer, fully connected layer (C)// only the number of
    class that are directly applied to softmax;
     Return (save as .csv file) extracted features
End
```

GLCM feature extraction

GLCM method is a way of extracting second order statistical texture features. It examines the spatial relationship among pixels and defines how frequently a combination of pixels are present in an image. As we mention in chapter 2 section 2.3.3 these are the five different texture features that GLCM offers. It can be gained by summing up all co-occurrences of grey scale values at a specified offset over an image, with those attributes. In this study we use those five features of GLCM to extract feature from fundus image.

GLCM + CNN feature

The merged features are made up of elements from both CNN and GLCM. In order to merge CNN and GLCM features, which are both in the form of rows while the latter are in matrices, we must apply a transpose matrix to the GLCM features, which converts the matrix form into rows. 517 characteristics total—512 from CNN (extracted features by CNN model that we describe it in 3.1 and 5 from GLCM—are used in our analysis.

To evaluate effectiveness of the proposed method, we made comparison into three scenarios. First, we evaluate CNN feature with (KNN, CNN and SVM) classifier the second scenario, we examine the performance of GLCM feature using (KNN and SVM) classifiers and lastly the performance of combined feature GLCM-CNN with KNN and SVM classifiers. The reason we combine both is in order to get the best feature of the optical image as a result of the energy attribute in GLCM is the best feature that is going to extract using GLCM not in the feature extraction of CNN.

3.6. Classification

Then to classify glaucoma from fundus image the extracted features are fed to three classification models, which are CNN + Softmax, SVM, and KNN. Because of its Simplicity to implement and intuitive to understand we choose KNN. It does not need training time for classification as it is a lazy learner and it is also only required one hyper-parameter k-value. When we come to SVM, it is a well-known traditional machine learning approach with it is best for binary classification, KNN. We select CNN as it is easy to understand and fast to implement. It has the highest accuracy among all algorithms that predicts images. We also select those three to evaluate their effect for our case in addition to those reasons.

Image classification with raw data, data with pre-processing, and pre-processed segmented data using the three classification models is applied. The steps in the classification process include:

Case I: CNN + Softmax

To find the probabilistic value, a flattening layer of CNN is used to convert all the resultant 2-Dimensional arrays from pooled feature maps into a single long continuous linear vector. Then the flattened matrix is fed as input to the fully connected layer dense layers containing a softmax activation function (for the highest likelihood classification) to classify the image.

Softmax: - is a mathematical function that converts a vector of numbers into a vector of probabilities, where the probabilities of each value are proportional to the relative scale of each value in the vector. The output from the previous CNN last fully connected layer is given to softmax classifier.

Algorithm 5: softmax classifier

```
Input: Extracted features
Output: Class label
begin
    Get the extracted feature;
    Apply the softmax classifier;
    Return the class label;
End
```

Case II: SVM

In the second case to use the SVM classifier over CNN features we removed the last dense layer which contains the SoftMax activation function and then the extracted features are saved as CNNfeatures1.csv file. Then the CSV file is inputted into the SVM classifier.

Case III: KNN

To read the CNN, GLCM and combined extracted features from a CSV file, we have used the loadtxt () method of the pandas library. Training and testing data sets were split using the 80:20 ratio for training and validation sets. The train_test_split method of Scikit-Learn is used to divide data into training and test sets.

Table 3.1: Description of the CNN model with a number of parameters

Model: "sequential"

Layer (type)	**Output Shape**	No of Parameters
conv2d (Conv2D)	(None, 1, 96, 16)	13840
max_pooling2d (MaxPooling2D)	(None, 1, 20, 16)	0
activation (Activation)	(None, 1, 20, 16)	0
conv2d_1 (Conv2D)	(None, 1, 20, 32)	4640
max_pooling2d_1(MaxPooling2D)	(None, 1, 7, 32)	0
activation_1 (Activation)	(None, 1, 7, 32)	0
conv2d_2 (Conv2D)	(None, 1, 7, 64)	51264
max_pooling2d_2 (MaxPooling2)	(None, 1, 3, 64)	0
activation_2 (Activation)	(None, 1, 3, 64)	0
conv2d_3 (Conv2D)	(None, 1, 3, 128)	73856
average_pooling2d(AveragePooling2)	(None, 1, 1, 128)	0
activation_3 (Activation)	(None, 1, 1, 128)	0
conv2d_4 (Conv2D)	(None, 1, 1, 256)	819456
average_pooling2d(AveragePooling2)	(None, 1, 1, 256)	0
activation_4 (Activation)	(None, 1, 1, 256)	0
conv2d_5 (Conv2D)	(None, 1, 3, 512)	1180160
average_pooling2d(AveragePooling2)	(None, 1, 1, 512)	0
activation_5 (Activation)	(None, 1, 1, 512)	0
dropout (Dropout)	(None, 1, 1, 512)	0
flatten (Flatten)	(None, 512)	0
The extracted CNN features	**For feature 512**	
Total params: 2,143,216 **Trainable params: 2,143,216** **Non-trainable params: 0**		
Classification begun using softmax		

dense (Dense)	(None, 512)	262656
dropout_1(Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 2)	1026
Total params: 2,406,898 **Trainable params: 2,406,898** **Non-trainable params: 0**		

3.7. Evaluation metrics

Different performance metrics have been employed to assess the efficacy of the proposed solution or model. Accuracy, precision, recall, and f1-score are among those that are frequently used to evaluate how well proposed methods perform. True positive, true negative, false positive, and false negative parameters must be known in order to determine the values of the aforementioned performance matrices.

- **True positive (TP):-** is an outcome where the model correctly predicts the positive class. TP model result is one that detects the glaucoma condition when the condition is present.
- **True negative (TN):-** is an outcome where the model correctly predicts the negative class. TN model result is one that does not detect the glaucoma condition when the condition is absent.
- **False positive (FP):-** is an outcome where the model incorrectly predicts the positive class. FP model result is one that detects the glaucoma condition when the condition is absent.
- **False Negative (FN):-** is an outcome where the model incorrectly predicts the negative class. FP model result is one that does not detect the glaucoma condition when the condition is present.

Accuracy: is calculated as the sum of correct classifications (include both true positives and true negatives) divided by the total number of classifications. Accuracy can be misleading if used with imbalanced datasets, and therefore there are other metrics which can be useful for evaluating performance. Mathematical expression is given below.

$$\textit{Accuracy}= \frac{T_P+T_N}{T_P+F_P+T_N+F_N} \tag{3.1}$$

Precision: is the measurement of the proportion of true positives against the whole positives. It is also known as the positive predictive value. Mathematically, it is expressed as:

$$\textit{Precision}= \frac{T_P}{T_P+F_P} \tag{3.2}$$

Recall: it is also known as the sensitivity of a model and It's a metric for determining how much the true positive is among all positives and negatives. Mathematically, it is expressed as:

$$\textit{Recall} = \frac{T_P}{T_P+F_N} \tag{3.3}$$

F1-score: is also known as F-value which uses both precision score and recall score of a classifier. Mathematically, it is expressed as:

$$\textit{F1-score} = \frac{2*(\text{Precision}*\text{ Recall})}{\text{Precision}+\text{Recall}} \tag{3.4}$$

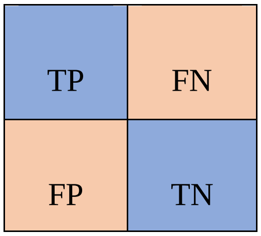
Micro Average: Summing up the individual true positives, false positives and false negatives for each class then micro precision, micro recall and micro F1Score is calculated using formula below

$$\textit{Micro\_Average\_precision} = \frac{T_{P1}+T_{P2}+\cdots+T_{PN}}{(T_{P1}+T_{P2}+\cdots+T_{PN})+(F_{P1}+F_{P2}+\cdots+F_{PN})} \tag{3.5}$$

$$\textit{Micro\_Average\_Recall} = \frac{T_{P1}+T_{P2}+\cdots+T_{PN}}{(T_{P1}+T_{P2}+\cdots+T_{PN})+(F_{N1}+F_{N2}+\cdots+F_{NN})} \tag{3.6}$$

$$\textit{Micro\_Average\_F}_1\textit{Score}= \frac{2*(\text{Micro\_precision} * \text{Micro\_Recall })}{\text{Micro\_precision} + \text{Micro\_Recall}} \tag{3.7}$$

Accuracy can be misleading if used with imbalanced datasets, and therefore there are other metrics based on confusion matrix which can be useful for evaluating performance. Also, a confusion matrix is used to show the prediction accuracy on the testing data set which means, to evaluate the quality of the model. Confusion matrix is a very popular measure used while solving classification problems. It can be applied to binary classification as well as for multiclass classification problems. Confusion matrices is a table that represent counts from predicted and actual values.(Kulkarni et al., 2020). For binary classification, the scheme of the confusion matrix table is as seen as in Figure 3.



TP	FN
FP	TN

Predicted class

Figure 3.9: description of confusion matrix

CHAPTER FOUR

RESULT AND DISCUSSION

4.1. Introduction

This chapter explained and presented the experimental evaluation of the proposed model based on the trained model of the system. The main objective of this work is to design and develop a model for detecting glaucoma from fundus images. The execution of the proposed models is approved by experimental assessment. The below section has a brief discussion of the dataset used, how the suggested model was put into practice, and the experimental results. In our model, the impact of pre-processing, data augmentation, and different classifiers are also assessed.

4.2. Dataset

For this study, glaucoma fundus images classified by ophthalmologists are taken up on request from publicly available LAG database. As we mention in chapter 3 section 3.2, the total number of the dataset is 11,750 with this number 4878 images belonging to positive glaucoma and 6882 belonging to negative glaucoma. However, the one that we found and which is freely available up on request was LAG database part one which contains a total of 4814 fundus images (i.e., 1711 glaucomatous images and 3143 healthy images). The fundus images on this dataset are taken by three different kinds of cameras: Topcon, Canon, and Carl Zeiss and all the images are in jpg (Joint Photographer Expert Group) file format and have a pixel size range from 582 x 597 to 3456 x 5184. As we see from the above description the number of glaucomatous images is less than that of the healthy ones so on the enhanced datasets, image augmentation was used only in glaucomatous images to balance the number of data sets in each class and improve the classification accuracy of the model. In this study, each image has a 96 x 96-pixel resolution (resized by bicubic interpolation) and is stored as a jpg file. We use a total of 6000 images and our data is divided into two categories: training and testing, with an 80:20 ratio: 4800 is used for model training and the remaining 1200 being used to evaluate the performance of the proposed model to determine how well it generalizes.

Table 4.1 Dataset description

Original LAG database format description					
NO	Class	Source	Resolution	Image format	Quantity
1	Healthy	LAG	Average of 1977 x 2594	Jpg	3143
2	Glaucomatous	LAG	Average of 1977 x 2594	Jpg	1711
LAG Dataset used in the proposed system format description					
1	Healthy	LAG	96 x 96	Jpg	3000
2	Glaucomatous	LAG	96 x 96	Jpg	3000

4.3. Implementation

In this section, the performance of the proposed models was evaluated using the data described in the previous section. The proposed model was developed using Keras (Tensor flow of python 3.9 as a backend). And anaconda Spyder IDE 5.1.5 is used on a Window 10 desktop computer with an Intel(R) Core (TM) i7-8700U CPU @ 3.19GHz and 8.00 GB RAM using the sci-kit-learn, Tensor flow, Keras, pillow, and OpenCV libraries, which use the python programming language. The model is trained for 100 epochs, a batch size of 32, and a starting or initial learning rate of 0.001 (1e-3). In CNN training, the initial learning rate is a crucial hyperparameter. Longer training times come from smaller values. For higher levels of learning rate, the model may diverge or fail to converge. Therefore, choosing the right initial learning rate is crucial. To determine the range of initial learning rate values that provide improved classification accuracy, simulation tests are carried out. The network is trained and evaluated using various initial learning rate values (1e-2 to 1e-9) and 0.001 (1e-3) is selected for the given dataset and also a default learning rate of adam optimizer in eras. The data is partitioned into a train, validating, and testing with an 80:10:10 ratio respectively.

4.4. Experimental Results

We used the precision, accuracy, recall, F1-score, micro average, weight average, and confusion matrix to evaluate the performance of our model. Additionally, we compare our model's performance to that of other related research studies, evaluate various feature extraction methods, and test both preprocessed and unprocessed datasets. We test various optimizers using various learning rates and epochs to see the impact of optimizers on the performance of end-to-end models.

4.4.1. Preprocessing

According to the research performed by (Hani et al., 2013) it is observed that retinal fundus images contained both additive noise (for e.g. Gaussian noise) and multiplicative noise (for e.g. speckle noise). To remove those noises, we use different noise removal techniques like anisotropic, bilateral filtering, BM3D, morphological closing, and opening median filtering after resizing it using bicubic interpolation for a better result. The PSNR value for BM3D, Anisotropic, median, and bilateral filtering are 47.20, 48.77, 52.32, and 48.27 respectively for a randomly picked image in our dataset. From those we choose median filters as a result of it has high PSNR for our dataset as compared to the other techniques as a result of SNR can reflect the magnified image quality, the larger SNR means the higher image quality(Han, 2013). We use the median filter with kernel size 3 X 3.and also as it is known for keeping edges sharp during noise removal, the median filter plays a key role in the field of image processing. The other preprocessing step we perform in this study after resizing and de-nosing is that histogram equalization. In this step, we choose CLAHE as a result of normal HE has a tendency to amplify noise than CLAHE even though we have been smoothed the image using edge preserving median filter. The other reason is that in fundus image the important part in the detection of Glaucoma is based on the bright circle inside the OD. So, we get better result when we use CLAHE than other histogram equalization technique since CLAHE limit the contrast while enhancing the image.

KNN classification

In this experiment, a pre-processed dataset with just augmented glaucoma is used to assess the model's performance because we have enough number of healthy image. In this experiment, a KNN algorithm with a K-value of 7 and an optimal parameter for Euclidean distance is used.

4.1.1. Classification using KNN with GLCM feature

The contrast, energy, homogeneity, dissimilarity, and entropy attributes from the hand-crafted GLCM texture-based feature extraction algorithm have been obtained and fed to the KNN classifier with a number of neighbors K=7. Table 4.2 shows the results for precision, accuracy, recall, F1-score, micro average, and weight average.

Table 4.2 GLCM features with KNN classifier Performance evaluation

GLCM feature by KNN classifier	**Class**	**Precision**	**Recall**	**Score**	**Accuracy**
	Glaucomatous	0.71	0.69	0.70	68.66%
	Healthy	0.66	0.68	0.67	
Micro average		0.69	0.69	0.69	
Weight average		0.69	0.69	0.69	

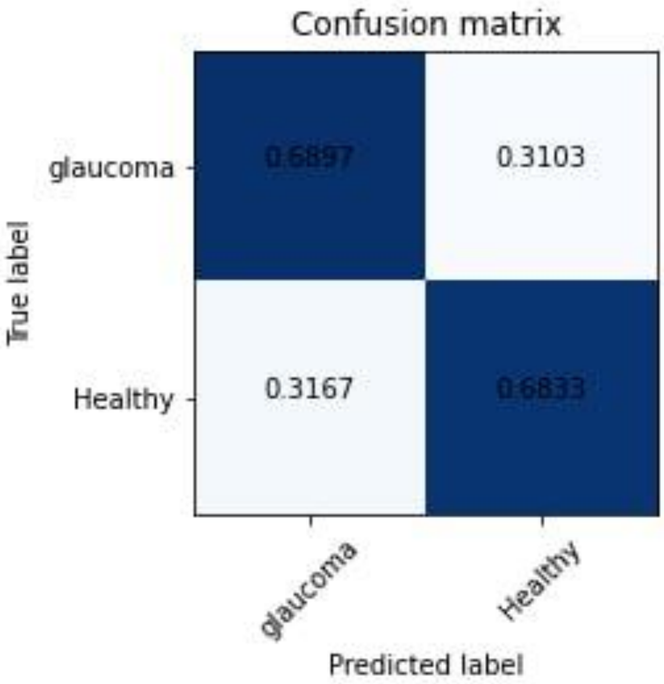


Figure 4.1: Confusion matrix for GLCM feature by KNN classifier

The above confusion matrix shows that the KNN classifier wrongly categorized 31.03% of glaucomatous image as healthy image and correctly categorized glaucomatous images as 68.97 % of the given dataset. In the other hand KNN classifier confused and categorized 31.67% of healthy image as glaucomatous image and as exactly categorized 68.97 % of the dataset as healthy as the true value. In this experiment identifying healthy achieve lower accuracy when compared with glaucomatous. This model achieves 68.66 % overall accuracy. This result shows that there is still valuable information need to be extracted in the fundus image especially in healthy image and we need better classifier to get a good accuracy.

4.1.2. Classification using KNN with CNN feature

In this experiment a feature extracted by CNN is employed to examine the impact on the KNN classifier model. CNN, a deep learning feature extraction algorithm, was used to extract 512 features, which were then given to the KNN classifier. Results for F1-score, Micro average, Weight average, and recall are displayed in Table 4.3.

Table 4.3 : CNN features with KNN classifier Performance evaluation

CNN feature by KNN classifier	Class	Precision	Recall	Score	Accuracy
	Glaucomatous	0.73	0.87	0.79	75.5%
	Healthy	0.81	0.63	0.71	
Micro average		0.77	0.75	0.75	
Weight average		0.76	0.76	0.75	

As above Table 4.3 an accuracy of 75% is gain for CNN features with KNN classifier. From this, we can clearly see that the average accuracy of features extracted by CNN is higher than on Hand-crafted features of GLCM when using KNN classifier.

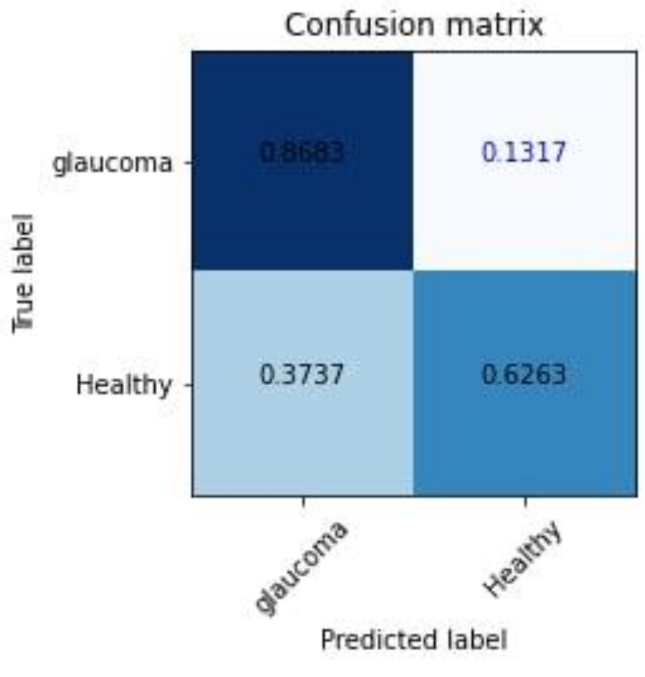


Figure 4.2: Confusion matrix for CNN feature by KNN classifier

The above confusion matrix shows that the KNN classifier wrongly categorized 13.17% of glaucomatous image as healthy image and correctly categorized glaucomatous images as 86.83%

of the dataset. In the other hand KNN classifier confused and categorized 37.37% of healthy image as glaucomatous image and as exactly categorized 62.97 % of the dataset as healthy as the true value. Still in this experiment identifying healthy achieve lower accuracy when compared with glaucomatous. This model achieves 75.5% overall accuracy which is better from the previous experiment. This result shows that the extraction and identification of the healthy image is less than that of the previous experimental result with GLCM feature even though it is improved on the glaucomatous image.

4.1.3. Classification using KNN with CNN and GLCM feature

Also in this experiment, the performance of KNN classifier is evaluated using combined features of GLCM and CNN with a total dimension of 517. The five attributes of handcrafted texture-based feature extraction. GLCM (Energy, correlation, homogeneity, Dissimilarity, and entropy) have been combined with the proposed CNN model's 512 features and provided to the KNN classifier with a total combined attribute of 517 features. We have done optimal parameter selection after experimenting the effect of each attribute by removing each one by one. Results for precision, accuracy, recall, F1-score, micro average, and weight average are displayed in Table 4.4.

Table 4.4: CNN+GLCM features with KNN classifier Performance evaluation

CNN+GLCM feature by KNN classifier	**Class**	**Precision**	**Recall**	**Score**	**Accuracy**
	Glaucomatous	0.77	0.93	0.84	81.66%
	Healthy	0.90	0.68	0.78	
Micro average		0.84	0.81	0.81	
Weight average		0.83	0.82	0.81	

As above Table 4.4 an accuracy of the 81% is gain for hybrid GLCM + CNN features with KNN classifier. From this we can clearly see that the average accuracy of features extracted by hybrid GLCM+CNN is higher than the previous experiments when using KNN classifier.

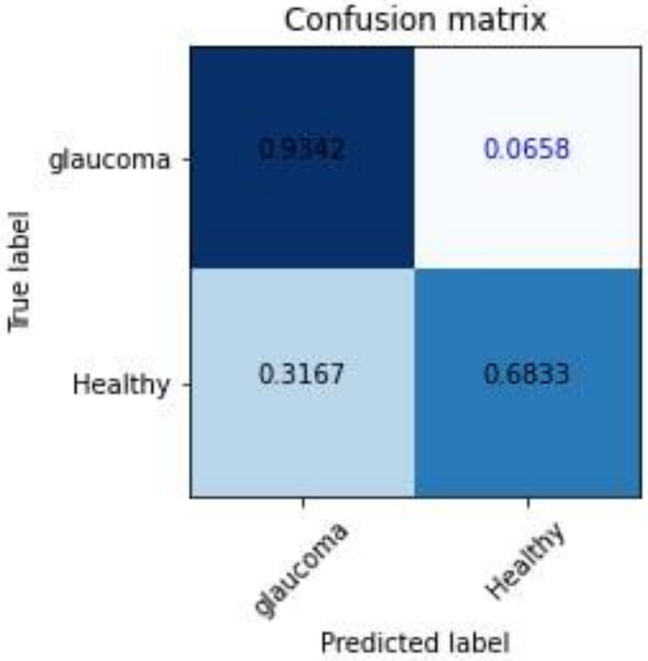


Figure 4.3: Confusion matrix for CNN+GLCM feature by KNN classifier

The above confusion matrix shows that the KNN classifier wrongly categorized 6.58% of glaucomatous image as healthy image and exactly categorized glaucomatous images as 93.42% of the dataset. In the other hand KNN classifier confused and categorized 31.67% of healthy image as glaucomatous image and as exactly categorized 68.33 % of the dataset as healthy as the true value. In still in this experiment identifying healthy achieve lower accuracy when compared with glaucomatous. This result shows that the identification of the glaucomatous image better that of healthy image each time. When we come to healthy image categorization it is the same as the previous GLCM with KNN result.

SVM classification

Different kernel functions are used by various SVM algorithms. There are various forms of these functions. For instance, linear, nonlinear, polynomial, sigmoid, and radial basis function (RBF). After conducting different experiments using those kernel functions, we choose and used linear Kernel after getting a better result as compare to other kernel function. And it is also one of the most common kernels to be used and mostly used when there are large number of Features in a particular Data Set. And from our experimental evaluation, we get faster Training time than the other Kernels when we use SVM with a Linear Kernel.

4.1.1. Classification using SVM with GLCM feature

The same feature that is extracted using GLCM as in the prior experiment is used in this experiment, with the classifier being changed to SVM. Precision, accuracy, recall, Micro average, Weight average and F1-score results are shown in Table 4.5.

Table 4.5: GLCM features with SVM classifier Performance evaluation

GLCM feature by SVM classifier	**Class**	**Precision**	**Recall**	**Score**	**Accuracy**
	Glaucomatous	0.57	0.75	0.65	58.08%
	Healthy	0.61	0.40	0.48	
Micro average		0.59	0.58	0.56	
Weight average		0.59	0.58	0.57	

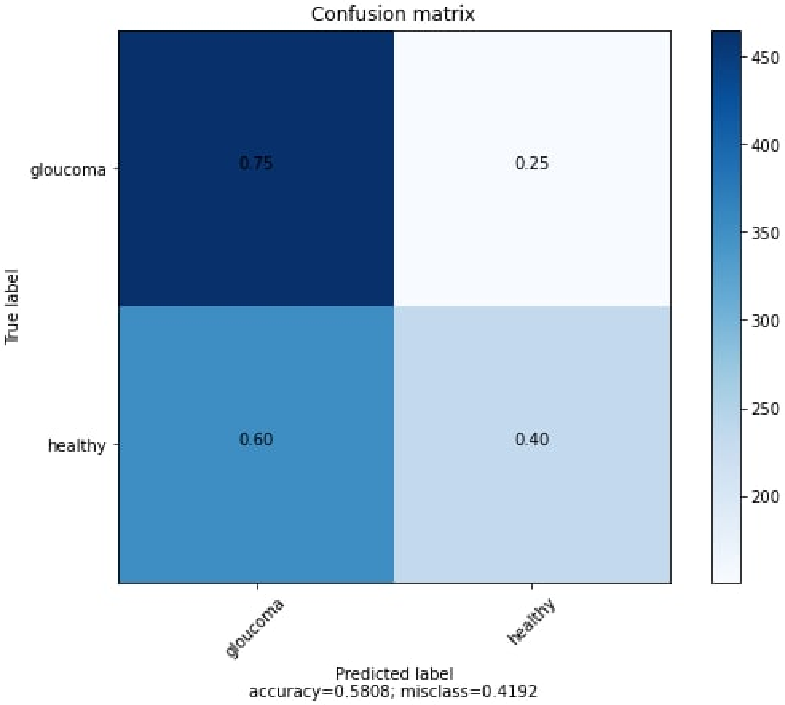


Figure 4.4: Confusion matrix for GLCM feature by SVM classifier

The above confusion matrix shows that the SVM classifier confused and categorized 25% of glaucomatous image as healthy image and exactly categorized glaucomatous images as 75 % of the dataset which is better than KNN with GLCM feature. In the other hand SVM classifier confused and categorized 60% of healthy image as glaucomatous image and as exactly categorized 40% of the dataset as healthy as the true value. In this experiment the prediction rate of healthy achieve very low accuracy when compared with glaucomatous and also as compared to the previous experiments. This model achieves 58.08 % overall accuracy which is the lowest result from the previous experiments.

4.1.2. Classification using SVM with CNN feature

The same feature that is extracted using CNN as in the prior experiment with 512 feature is used in this experiment, with the classifier being changed to SVM. Precision, accuracy, recall, Micro average, Weight average and F1-score results are shown in Table 4.6.

Table 4.6: CNN features with SVM classifier Performance evaluation

CNN feature by SVM classifier	**Class**	**Precision**	**Recall**	**Score**	**Accuracy**
	Glaucomatous	0.89	0.81	0.85	85.42%
	Healthy	0.82	0.90	0.86	
Micro average		0 86	0.86	0.85	
Weight average		0.86	0.85	0.85	

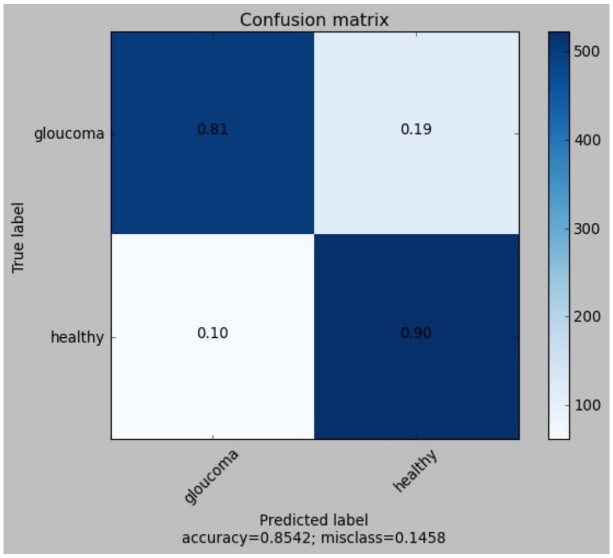


Figure 4.5: Confusion matrix for CNN feature by SVM classifier

The above confusion matrix shows that the SVM classifier confused and categorized 19% of glaucomatous image as healthy image and exactly categorized glaucomatous images as 81 % of the dataset which is less good than KNN with CNN feature. In the other hand SVM classifier confused and categorized 10% of healthy image as glaucomatous image and as exactly categorized 90% of the dataset as healthy as the true value. In this experiment the prediction rate of healthy got improved and achieve high accuracy when compared with glaucomatous and also as compared to the previous experiments. This shows that SVM classifier with linear kernel function become well when there is large number of Features in a particular dataset. This model achieves 85.42 % overall accuracy which is good result from the previous experiments.

4.1.3. SVM classification with extracted feature by CNN+GLCM

The same hybrid feature that is extracted using CNN and GLCM as in the prior experiment with 517 feature is used in this experiment, with the classifier being changed to SVM. Precision, accuracy, recall, Micro average, Weight average and F1-score results are shown in Table 4.7.

Table 4.7: CNN+GLCM features with SVM classifier Performance evaluation

CNN+GLCM feature by SVM classifier	**Class**	**Precision**	**Recall**	**Score**	**Accuracy**
	Glaucomatous	0.87	0.84	0.85	85.85%
	Healthy	0.85	0.88	0.86	
Micro average		0.86	0.86	0.86	
Weight average		0.86	0.86	0.86	

As we can see in the from confusion matrix on Figure 4.6, it shows that the SVM classifier confused and categorized 16% of glaucomatous image as healthy image and exactly categorized glaucomatous images as 84 % of the dataset which is less good than KNN with CNN + GLCM feature. In the other hand SVM classifier confused and categorized 12% of healthy image as glaucomatous image and as exactly categorized 88% of the dataset as healthy as the true value. In this experiment the prediction rate of healthy got improved and achieve high accuracy when compared with glaucomatous and also as compared to the previous KNN classification experiments with the same combined feature. This shows that SVM classifier become even well when there is large number of Features in a particular dataset when using a linear kernel function. This model achieves 85.85 % overall accuracy which is a little bit higher from the previous experiment and got higher than previous experiments. Precision, accuracy, recall, Micro average, Weight average and F1-score results are shown in Table 4.7.

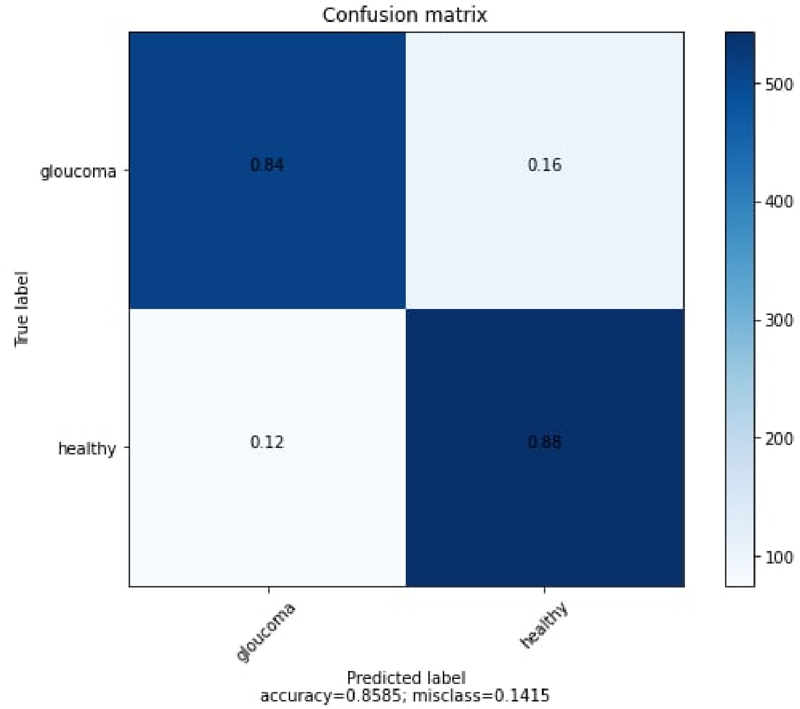


Figure 4.6: Confusion matrix for CNN+GLCM feature by SVM classifier

4.1.4. End to end CNN with different optimizers

End-to-end CNN model Performance with Adam optimizer

The performance of the proposed model with Adam optimizer is evaluated to see the effect of Adam optimizer in the proposed model. Precision, accuracy, recall, Micro average, Weight average and F1-score results are shown in Table 4.8.

Table 4.8: End-to-end CNN model with Adam optimizer Performance evaluation

End to end CNN with Adam optimizer	Class	Precision	Recall	Score	Accuracy
	Glaucomatous	0.94	0.92	0.93	93%
	Healthy	0.91	0.94	0.93	
Micro average		0.93	0.93	0. 93	
Weight average		0.93	0.93	0.93	
Training accuracy 98.40% **Validation accuracy 93.70%** **Test accuracy 93%**					

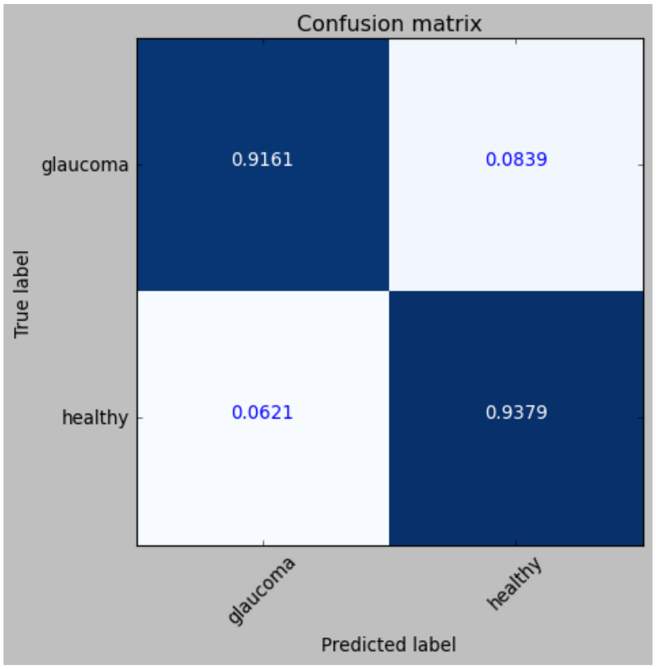


Figure 4.7: End-to-end CNN model with Adam optimizer

The confusion matrix on the above figure shows that the result of end-to-end CNN. The softmax classifier of CNN confused and categorized 8% of glaucomatous image as healthy image and exactly categorized glaucomatous images as 91 % of the dataset which is less than KNN with CNN + GLCM feature. In the other hand the classifier confused and categorized 6% of healthy image as glaucomatous image and as exactly categorized 93% of the dataset as healthy as the true value which is the best result from the previous experiments. In this experiment the prediction rate of healthy got improved and achieve high accuracy when compared with glaucomatous and also as compared to the previous all classification experiments with the same CNN proposed model. This shows that deep learning has best performance with CNN feature extraction and softmax classifier. This model achieves 93 % overall accuracy which is a higher from the previous experiments.

End-to-end CNN model Performance with RMSprop optimizer

RMSprop optimizer is evaluated to see the effect of the RMSprop optimizer in the proposed model. Precision, accuracy, recall, Micro average, Weight average and F1-score results are shown in Table 4.9.

Table 4.9: End-to-end CNN model with RMSprop optimizer Performance evaluation

End to end CNN	Class	Precision	Recall	F1-Score	Accuracy
	Glaucomatous	0.90	0.94	0.92	92%
	Healthy	0.93	0.89	0.91	
Micro average		0.92	0.92	0.92	
Weight average		0.92	0.92	0.92	

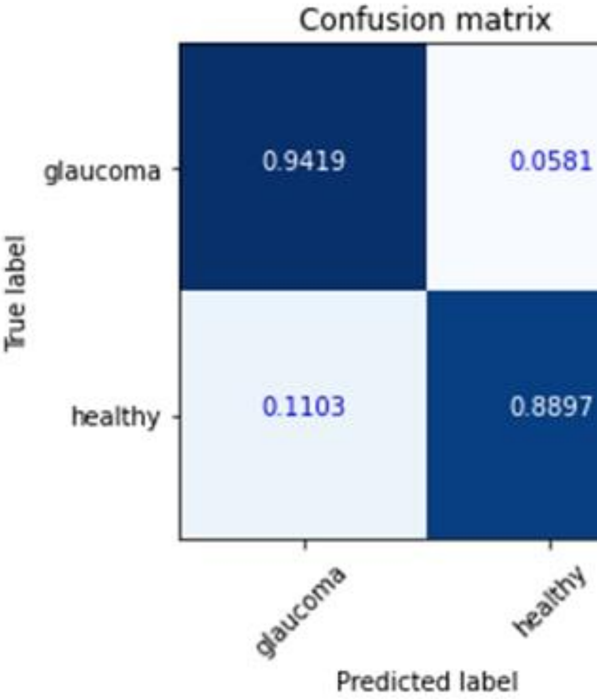


Figure 4.8: End-to-end CNN model with RMSprop optimizer

As we can see in the above confusion matrix, when we use RMSprop we get less accuracy than Adam optimizer. This model achieves 92% accuracy

End-to-end CNN model Performance with SGD optimizer

Proposed model with SGD optimizer is evaluated to see the effect of the SGD optimizer in the end-to-end CNN model. Precision, accuracy, recall, Micro average, Weight average and F1-score results are shown in Table 4.10.

Table 4.10: End-to-end CNN model with SGD optimizer Performance evaluation

	Class	Precision	Recall	F1-Score	Accuracy
End to end CNN SGD optimizer	Glaucomatous	0.89	0.92	0.90	90%
	Healthy	0.91	0.88	0.89	
Micro average		0.90	0. 90	0.90	
Weight average		0.90	0.90	0.90	

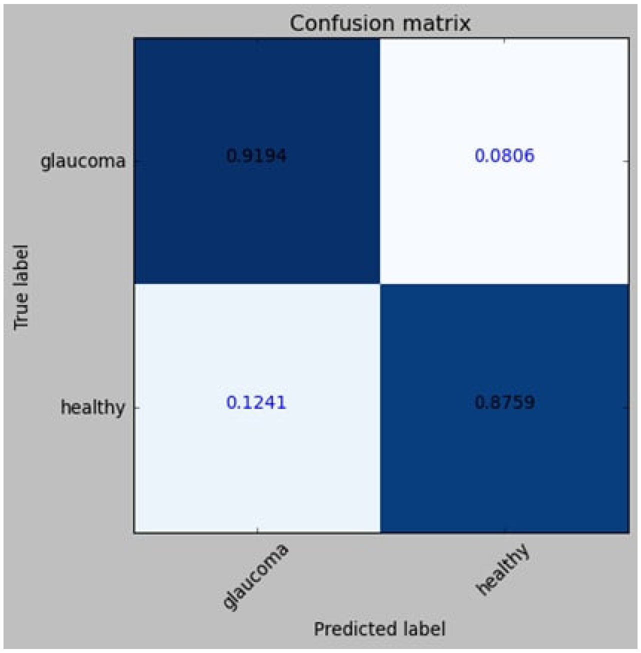


Figure 4.9: End-to-end CNN model with SGD optimizer

As we can see in the above confusion matrix, when we use SGD even more, we get less accuracy than Adam and RMSprop optimizer. This model achieves 90% accuracy. Therefore, we use Adam optimizer as a result of its performance on our model for the given dataset.

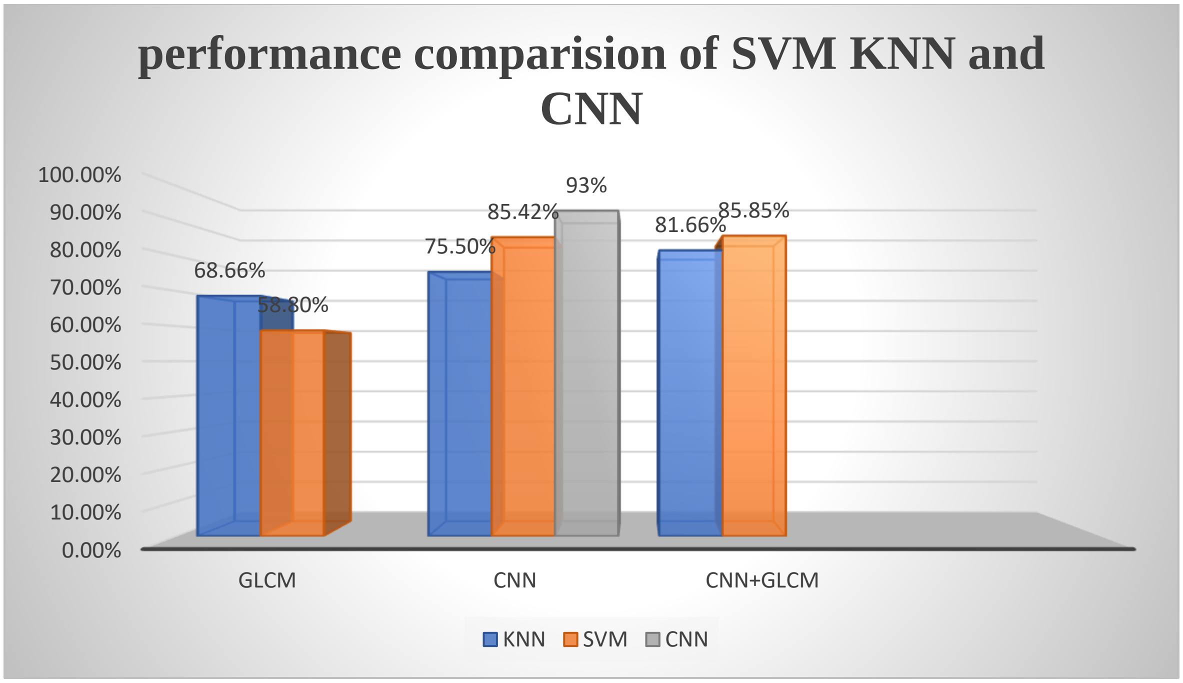


Figure 4.10: Comparison of SVM, KNN and CNN classifier

End to end CNN Model Performance with Noisy Dataset and without augmentation

In this experiment, the proposed model's performance is assessed using a noisy dataset to determine the impact of various preprocessing techniques. We use the Fundus image dataset for this experiment, with no preprocessing other than resizing using the same CNN model with adam optimizer. Precision, accuracy, recall, Micro average, Weight average and F1-score results are shown in Table 4.11.

Table 4.11: Performance evaluation of end-to-end CNN model with noisy datasets

End to end CNN With noisy dataset Adam optimizer	Class	Precision	Recall	F1-Score	Accuracy
	Glaucomatous	0.84	0.91	0.87	87%
	Healthy	0.91	0.84	0.87	
Micro average		0.88	0.87	0.87	
Weight average		0.88	0.87	0.87	

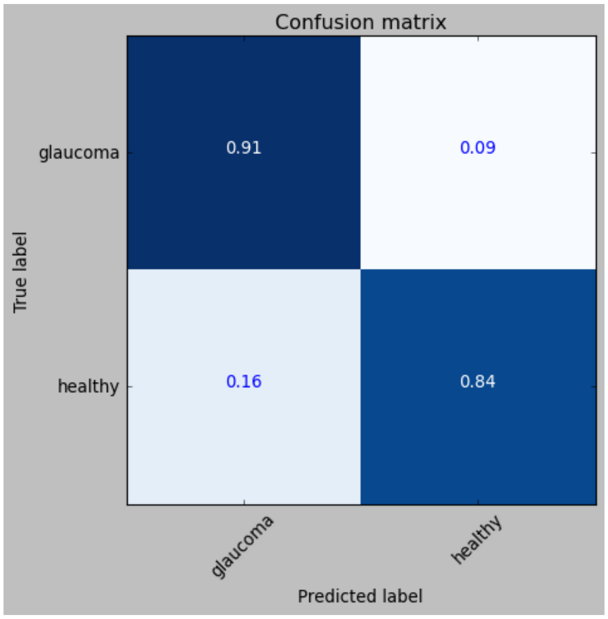


Figure 4.11: End-to-end CNN model with CNN model with noisy dataset

The above confusion matrix clearly shows that the 9% of noisy glaucomatous images are categorized as healthy image and the other 91% are other exactly categorized glaucomatous images. When we come to healthy dataset the 16% of noisy healthy images are categorized as glaucomatous image and the other 84% are other exactly categorized healthy images. The overall accuracy that we get from this experiment is 87%. As the experimental result indicate the accuracy is very low than preprocessed dataset accuracy, this indicates the model with noisy dataset have low ability to generalize.

Table 4.12: Performance comparison of the before preprocessing and after preprocessing

Proposed model CNN	**Accuracy Before preprocessing**		**Accuracy After preprocessing**	
Glaucoma TP	91%	87%	91.61%	93%
Healthy TN	84%		93.79%	

Table 4.13: Performance comparison of the proposed model and related works

No.	**References**	**dataset**	**techniques**	**Optimizer**	**Result**
1	**(Elangovan & Nath, 2021)**	DRISHTI–GS1,	preprocessing (Only image resizing and augmentation), feature extraction and	Adam	*94.43%* On LAG

		ORIGA, RIMONE2 ACRIMA *LAG*	classification (self-build CNN)		
2	**(Raghavendra et al., 2018)** **CNN**	Dataset prepared by them	preprocessing (Only image resizing), feature extraction and classification (self-build CNN)	Not mentioned	98%
3	**(Saxena et al., 2020)** **CNN**	ORIGA SCES	Only ROI extraction and augmentation, feature extraction and classification (self-build CNN)	Not mentioned	AUC 0.0822 0.882
4	**(Sallam et al., 2021)**	LAG	Only augmentation and AlexNet, VGG11, VGG16, VGG19, GoogleNet, ResNET-18, ResNET-50, ResNET-101 and ResNet-152	------	86.9% ResNet-152
5	(**Afroze et al., 2021**)	Combined ACRIMA and LAG dataset	Only augmentation and feature extraction (Inception V3) classification (proposed CNN)	Not mentioned	85.29 %
6	**(Vaghjiani et al., 2020)**	ACRIMA, Drishti- GS, HRF, RIM-ONE r2, sjchoi86 HRF and DRIONS-DB	No preprocessing, feature extraction and classification (VGG16)	------	93%
7	**Our model**	LAG	Preprocessing (Bicubic resize, median filtering, CLAHE), augmentation, feature extraction and classification (self-build CNN)	Adam with learning rate 0.001	**93%**

4.5. Answer to Research Question

Research Question 1: What effective image-processing technique should be applied to improve the fundus image?

The experimental results show that the proposed model achieves an overall accuracy of 87 percent with a downsized dataset and no additional preprocessing. As we can see from table 4.12 there is high confusion between healthy with glaucoma and the model categorized about 16% of the healthy test datasets as glaucomatous as a result of the bright area inside the OD is not clearly placed and confuses the model. On the other hand, the proposed model using the preprocessed dataset improves the detection of healthy images from 84 to 93 percent, yielding an overall test accuracy of 93 percent. The proposed model uses a bicubic interpolation for resizing the image dataset, median filter to remove noise from the dataset and CLAHE to improve contrast. According to the experimental findings, techniques for resizing, noise reduction, and contrast enhancement are preferred in the pre-processing stage, especially to enhance the fundus image and improve the probability of detection prediction of healthy image as a true label. Therefore, the noise reduction and contrast enhancement boost the suggested model's overall performance by 6%, especially the healthy image's TN rate, which jumps from 84 to 93.79 percent by 9.79%.

Research Question 2: To what extent preprocessing increase the performance of CNN model?

We have conducted a variety of tasks to enhance the model's performance. Due to the imbalance of our dataset, data augmentation has been done. After experimenting with the value of accuracy and computational time for a range of 32 to 224 randomly chosen values, the dataset's dimension is also resized to 96x96. After testing various filters, we choose the median filter since it produces the highest PSNR results. CLAHE is then used to enhance contrast. Different classification strategies are used to improve the detection process, either separately or in combination using CNN and GLCM features, however the develop CNN model with Adam optimizer achieves the best results for glaucoma detection using fundus images.

Research Question 3: Which optimizer achieve the best in our model and dataset?

In experiment one we evaluate the performance of the proposed model with the three optimizers. From the optimizers we use Adamax achieve the best performance. In other hand SGD achieve the list accuracy of all the three optimizers we have tested.

Research Question 4: To what extent the identification accuracy is registered?

We employed quantitative performance analysis, which includes precision, accuracy, recall, micro average, weight average, and F1-score, to evaluate the model's performance. In addition to that, to see in detail description of the classification rate vs. misclassification rate, confusion matrix is employed. As shown by the experimental results in figure 4.11, when compared to other approaches that we perform, the proposed CNN model with adam optimizer obtains the highest accuracy for the detection of glaucoma using fundus images. The proposed model performs better and obtains Training accuracy 98.40%, Validation accuracy 93.70% and Test accuracy 93%.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1. Conclusion

Glaucoma is complication that is associated with the damage of optic nerve and causes permanent blindness. Glaucoma is one of the leading causes of blindness worldwide and also in our country. It is also known as the “silent thief” of sight since symptoms don't show up until the condition is fairly advanced. Even though glaucoma cannot be cured, therapy can halt its progression. Therefore, early glaucoma detection will be essential for a quick diagnosis. Effective imaging is crucial for the early identification of glaucoma. There are five factors should be checked before making a glaucoma diagnosis: Regular glaucoma check-ups include two routine eye tests: tonometry and ophthalmoscopy which is time taking. currently one of the most common and important methods for diagnosing glaucoma is by using digital fundus image which need the supervision of an expert to detect glaucoma from this image. Therefore, computer aided automatic detection of glaucoma from fundus image helps the expert ophthalmologist to detect the diseases early as possible and to take a proper diagnosis.

We propose a model in order to detect glaucoma from fundus image. To do the work we follow steps like pre-processing, data augmentation, feature extraction and classification. The pre-processing includes resizing by bi-cubic interpolation and removal of noise by using median filter. For data augmentation we use image data generator and we have used CNN for feature extraction and classification. The proposed CNN model contains 6 convolutional layers.

In our research we done different experiments. From this research preprocessing techniques such as bi-cubic interpolation technique, median filter and CLAHE are very effective for glaucoma detection as the result of the experiments indicate. The proposed model with a preprocessed dataset of LAG, with data augmentation and adam optimizer achieve the best result. The proposed model achieves 93% test accuracy with a batch size of 32 for 100 epochs.

5.2. Recommendation

The prime objective of the study was accomplished because the suggested model can address all research question and produce the desired results. Therefore, even though several issues still require more investigation, we can say that the model as proposed meets the current requirement. This research work can be further improved or implemented to identify glaucoma detection. We suggest the following issues for further research in this field based on the findings and analysis of this study.

- We gather the publicly accessible LAG database dataset for our research upon request. We advise gathering additional datasets from various databases for future work so that the effects can be observed.
- CNN was used for feature extraction, but to compare the model's performance, we advise adopting a combined technique of feature extraction.
- We test various optimizers to evaluate how their default learning rates affect the model's performance. We advise testing model performance by altering the learning rate for each optimizer other than Adam because the learning rate has its own effects.

References

Abràmoff, M., & Kay, C. N. (2012). Image Processing. *Retina Fifth Edition*, *1*, 151–176. https://doi.org/10.1016/B978-1-4557-0737-9.00006-0

Afroze, T., Akther, S., Chowdhury, M. A., Hossain, E., Hossain, M. S., & Andersson, K. (2021). Glaucoma Detection Using Inception Convolutional Neural Network V3. *Communications in Computer and Information Science*, *1435*, 17–28. https://doi.org/10.1007/978-3-030-82269-9_2

Bourne, R., Abdull, M., Pilippin, H., & Philippin, H. (2012). Figure 3. Glaucomatous optic neuropathy: splinter haemorrhages. In *Community EyE HEaltH Journal* | (Vol. 25, Issue 79).

Bourne, R. R. A., & Khatib, T. (2021). The optic nerve head in glaucoma. *Community Eye Health Journal*, *34*(112), 36–39.

Chitradevi, B., & Srimanthi, P. (2014). An Overview on Image Processing Techniques. *ISRN Signal Processing*, *2*(11), 6466–6472. http://www.hindawi.com/isrn/sp/2013/496701/

Davob, K., Foi, A., Katkovnik, V., & Egiazarian, K. (2007). The Development And Characterization Of Next Generation Endovascular Devices Using Thin Film Nitinol Notes The Development And Characterization Of Next Generation Endovascular Devices Using Thin Film Nitinol Notes. *IEEE Transactions on Image Processing*, *16*(8), 1–2.

DOREEN FAZIO. (n.d.). *What are the Symptoms of Glaucoma? | glaucoma.org*. Retrieved June 21, 2022, from https://glaucoma.org/what-are-the-symptoms-of-glaucoma/

Elangovan, P., & Nath, M. K. (2021). Glaucoma assessment from color fundus images using convolutional neural network. *International Journal of Imaging Systems and Technology*, *31*(2), 955–971. https://doi.org/10.1002/ima.22494

Exam, A. C. G. (2014). *Five Common Glaucoma Tests*. https://glaucoma.org/five-common-glaucoma-tests/

Fadnavis, S. (2014). Image Interpolation Techniques in Digital Image Processing: An Overview. *Journal of Engineering Research and Applications Www.Ijera.Com*, *4*(March), 70–73. www.ijera.com

Fan, L., Zhang, F., Fan, H., & Zhang, C. (2019). *S42492-019-0016-7.Pdf*. *7*.

Galgotias College of Engineering and Technology, IEEE Communications Society, IEEE Systems, M., & Institute of Electrical and Electronics Engineers. (2014). *Proceedings of the 2014 International*

Conference on Advances in Computing, Communications and Informatics (ICACCI) : 24-27 September 2014, Delhi, India. 2392–2397.

Ghosh, A., Sufian, A., Sultana, F., Chakrabarti, A., & De, D. (2019). Fundamental concepts of convolutional neural network. *Intelligent Systems Reference Library*, *172*, 519–567. https://doi.org/10.1007/978-3-030-32644-9_36

Guidelines, T., Glaucoma, E., & Foundation, S. (2017). European Glaucoma Society Terminology and Guidelines for Glaucoma, 4th Edition - Chapter 2: Classification and terminologySupported by the EGS Foundation. *British Journal of Ophthalmology*, *101*(5), 73–127. https://doi.org/10.1136/bjophthalmol-2016-egsguideline.002

Han, D. (2013). *Comparison of Commonly Used Image Interpolation Methods*. *Iccsee*, 1556–1559. https://doi.org/10.2991/iccsee.2013.391

Hani, A. F. M., Soomro, T. A., Fayee, I., Kamel, N., & Yahya, N. (2013). Identification of noise in the fundus images. *Proceedings - 2013 IEEE International Conference on Control System, Computing and Engineering, ICCSCE 2013*, 191–196. https://doi.org/10.1109/ICCSCE.2013.6719957

He, R., Luo, S., Jing, Z., & Fan, Y. (2011). Adjustable weighting image contrast enhancement algorithm and its implementation. *Proceedings of the 2011 6th IEEE Conference on Industrial Electronics and Applications, ICIEA 2011*, 1750–1754. https://doi.org/10.1109/ICIEA.2011.5975875

Horning, N. (2010). Random Forests: An algorithm for image classification and generation of continuous fields data sets. *International Conference on Geoinformatics for Spatial Infrastructure Development in Earth and Allied Sciences 2010*, 1–6.

Jadhav, S. D., & Channe, H. P. (2016). Comparative Study of K-NN, Naive Bayes and Decision Tree Classification Techniques. *International Journal of Science and Research (IJSR)*, *5*(1), 1842–1845. https://doi.org/10.21275/v5i1.nov153131

Jain, M., & Tomar, P. S. (2013). Review of image classification methods and techniques. *International Journal Of Engineering Research and Technology*, *2*(8), 852–858.

Jaiswal, S. (2015). *K-Nearest Neighbor(KNN) Algorithm for Machine Learning*. JavaTPoint. https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning

Khanum, M., Mahboob, T., Imtiaz, W., Abdul Ghafoor, H., & Sehar, R. (2015). A Survey on Unsupervised Machine Learning Algorithms for Automation, Classification and Maintenance. *International Journal of Computer Applications*, *119*(13), 34–39. https://doi.org/10.5120/21131-

4058

Kim, G., & Lee, C. (2016). Convolutional Neural Network Using Convolutional Neural Network. In *Springer* (Vol. 2644, Issue 2). Springer International Publishing. https://doi.org/10.1007/978-3-030-82269-9

Kolb, H., Ralph, N., & Eduardo, F. (1995). Webvision: The Organization of the Retina and Visual System [Internet]. *Salt Lake City (UT): University of Utah Health Sciences Center*, 1827.

Kotkar, V. (2013). REVIEW OF VARIOUS IMAGE CONTRAST ENHANCEMENT Related papers REVIEW OF VARIOUS IMAGE CONTRAST ENHANCEMENT. *International Journal of Innovative Research in Science, Engineering and Technology*, *2*(7).

Kulkarni, A., Chong, D., & Batarseh, F. A. (2020). Foundations of data imbalance and solutions for a data democracy. In *Data Democracy: At the Nexus of Artificial Intelligence, Software Development, and Knowledge Engineering*. Elsevier Inc. https://doi.org/10.1016/B978-0-12-818366-3.00005-8

Kumar, A. (2022). *Different Types of CNN Architectures Explained: Examples - Data Analytics*. https://vitalflux.com/different-types-of-cnn-architectures-explained-examples/

Kumar, R. (2012). FEATURE EXTRACTION OF DISEASED LEAF IMAGES Journal of Signal and Image Processing. *International Journal of Computer Science and Telecommunications*, *3*(1), 65–71. http://www.bioinfo.in/contents.php?id=48

Kwak, N. (2008). Feature extraction for classification problems and its application to face recognition. *Pattern Recognition*, *41*(5), 1701–1717. https://doi.org/10.1016/j.patcog.2007.10.012

Law, S. K., & Coleman, A. L. (2016). *Pearls of glau coma*.

Li, W., Mao, K., Zhang, H., & Chai, T. (2010). SELECTION OF GABOR FILTERS FOR IMPROVED TEXTURE FEATURE EXTRACTION. *Proceedings of 2010 IEEE 17th International Conference on Image Processing*, 361–364.

Magudeeswaran, V., & Singh, J. F. (2017). Contrast limited fuzzy adaptive histogram equalization for enhancement of brain images. *International Journal of Imaging Systems and Technology*, *27*(1), 98–103. https://doi.org/10.1002/ima.22214

Mayank Mishra. (2020). *Convolutional Neural Networks, Explained | by Mayank Mishra | Towards Data Science*. Towards Data Science. https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939

Ministry, F. (2003). *Federal Ministry of Health Ethiopia*. 1–59.

Mutlag, W. K., Ali, S. K., Aydam, Z. M., & Taher, B. H. (2020). Feature Extraction Methods: A Review. *Journal of Physics: Conference Series*, *1591*(1). https://doi.org/10.1088/1742-6596/1591/1/012028

National Glaucoma Research. (2021). *The Eye Exam for Glaucoma: 6 Common Tests | BrightFocus Foundation*. National Glaucoma Research. https://www.brightfocus.org/glaucoma/article/eye-exam-glaucoma

PADZIL, F. M. (2016). *Linear and Nonlinear Filter for Image Processing Using Matlab'S Image Processing Toolbox*. 1–65.

Palma, C. A., Cappabianco, F. A. M., Ide, J. S., & Miranda, P. A. V. (2014). Anisotropic diffusion filtering operation and limitations - Magnetic resonance imaging evaluation. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, *19*, 3887–3892. https://doi.org/10.3182/20140824-6-za-1003.02347

Patel, S., Bharath, K. P., Balaji, S., & Muthu, R. K. (2020). Comparative Study on Histogram Equalization Techniques for Medical Image Enhancement. *Advances in Intelligent Systems and Computing*, *1048*, 657–669. https://doi.org/10.1007/978-981-15-0035-0_54

Pinto, A. D., Morales, S., Naranjo, V., Köhler, T., Mossi, J. M., & Navea, A. (2019). CNNs for automatic glaucoma assessment using fundus images : an extensive validation. *BioMedical Engineering OnLine*, 1–20. https://doi.org/10.1186/s12938-019-0649-y

Priyanka C. Dighe, S. K. G. (2014). Survey on Image Resizing Techniques. *International Journal of Science and Research (IJSR)*, *3*(12), 1444–1448. https://www.ijsr.net/archive/v3i12/U1VCMTQ3MjI=.pdf

Raghavendra, U., Fujita, H., Bhandary, S. V, & Gudigar, A. (2018). Deep convolution neural network for accurate diagnosis of glaucoma using digital fundus images. *Information Sciences*, *441*, 41–49. https://doi.org/10.1016/j.ins.2018.01.051

Rashid, M. (n.d.). *Pseudocode of Decision Tree Algorithm | Download Scientific Diagram*. Retrieved July 16, 2022, from https://www.researchgate.net/figure/Pseudocode-of-Decision-Tree-Algorithm_fig1_338528758

Reza, M. S., & Ma, J. (2016). ICA and PCA integrated feature extraction for classification. *International Conference on Signal Processing Proceedings, ICSP*, *0*, 1083–1088. https://doi.org/10.1109/ICSP.2016.7877996

Sallam, A., Technology, I., Kaid, H. A. S., Technology, I., Technology, I., Gaid, A. S. A., Technology, I., Abdulkareem, R. A., Technology, I., Radman, A., Technology, I., Saif, W. Q. A., Technology, I., Ahmed, K. J. A., & Technology, I. (2021). *Learning from Pre-trained CNN Models*.

Saxena, A., Vyas, A., Parashar, L., & Singh, U. (2020). A Glaucoma Detection using Convolutional Neural Network. *Proceedings of the International Conference on Electronics and Sustainable Communication Systems, ICESC 2020*, *Icesc*, 815–820. https://doi.org/10.1109/ICESC48915.2020.9155930

Sinha, A., kumar, M., A.K, J., & Saxena, R. (2014). Performance Analysis of High Resolution Images Using Interpolation Techniques in Multimedia Communication System. *Signal & Image Processing : An International Journal*, *5*(2), 39–49. https://doi.org/10.5121/sipij.2014.5204

Sootla, S. (2015). Artificial neural network for image classification. *University of Tartu*, 15.

Stark, J. A. (2000). Adaptive image contrast enhancement using generalizations of histogram equalization. *IEEE Transactions on Image Processing*, *9*(5), 889–896. https://doi.org/10.1109/83.841534

Vaghjiani, D., Saha, S., Connan, Y., Frost, S., & Kanagasingam, Y. (2020). Visualizing and Understanding Inherent Image Features in CNN-based Glaucoma Detection. *2020 Digital Image Computing: Techniques and Applications, DICTA 2020*, 15–17. https://doi.org/10.1109/DICTA51227.2020.9363369

Vision, M. (1993). *Image pre-processing*. 56–57.

Vlachokosta, A. A., Asvestas, P. A., Matsopoulos, G. K., Uzunoglu, N., & Zeyen, T. G. (2007). Preliminary study on the association of vessel diameter variation and glaucoma. *Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings*, 888–891. https://doi.org/10.1109/IEMBS.2007.4352433

WHO. (2019). World report on vision. *World Health Organisation*, *214*(14), 180–235. https://www.who.int/publications-detail/world-report-on-vision

Win, N. N., Kyaw, K. K. K., Win, T. Z., & Aung, P. P. (2019). Image Noise Reduction Using Linear and Nonlinear Filtering Techniques. *International Journal of Scientific and Research Publications (IJSRP)*, *9*(8), p92113. https://doi.org/10.29322/IJSRP.9.08.2019.P92113

Wong, W. C. K., & Chung, A. C. S. (2004). A nonlinear and non-iterative noise reduction technique for medical images: Concept and methods comparison. *International Congress Series*, *1268*(C), 171–

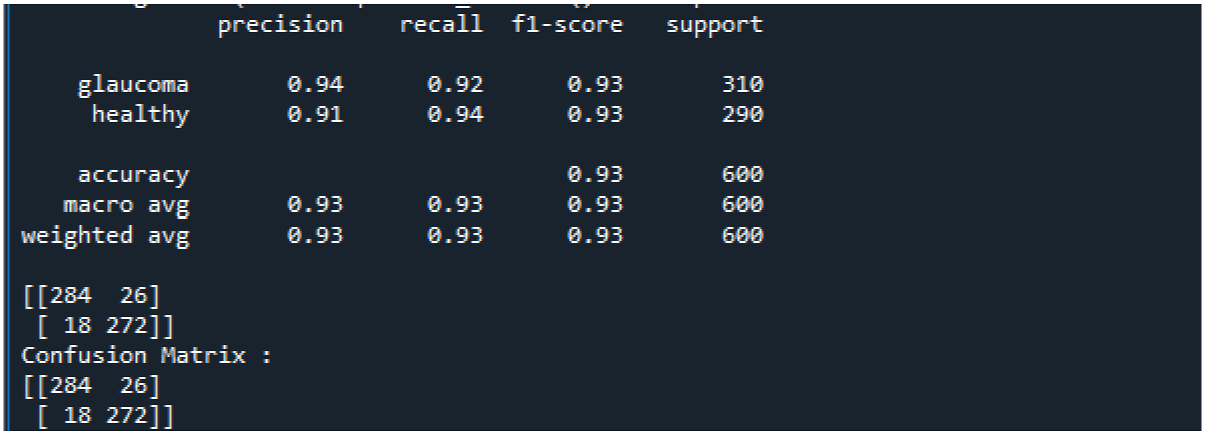
176. https://doi.org/10.1016/j.ics.2004.03.143

WSPOS ethiopia. (n.d.). *WSPOS | World Society of Paediatric Ophthalmology and StrabismusWSPOS Myopia Consensus Statement - WSPOS | World Society of Paediatric Ophthalmology and Strabismus*. Retrieved June 24, 2022, from https://www.wspos.org/ethiopia/

Wubet, G. M., & Assefa, A. A. (2021). Glaucoma and its predictors among adult patients attending ophthalmic outpatient department: a hospital-based study, North West Ethiopia. *BMC Ophthalmology*, *21*(1), 1–9. https://doi.org/10.1186/s12886-021-02168-y

Yaqub, M., Jinchao, F., Zia, M. S., Arshid, K., Jia, K., Rehman, Z. U., & Mehmood, A. (2020). State-of-the-art CNN optimizer for brain tumor segmentation in magnetic resonance images. *Brain Sciences*, *10*(7), 1–19. https://doi.org/10.3390/brainsci10070427

Appendices A



```
              precision    recall  f1-score   support

    glaucoma       0.94      0.92      0.93       310
     healthy       0.91      0.94      0.93       290

    accuracy                           0.93       600
   macro avg       0.93      0.93      0.93       600
weighted avg       0.93      0.93      0.93       600

[[284  26]
 [ 18 272]]
Confusion Matrix :
[[284  26]
 [ 18 272]]
```

Python source code for proposed model

Python source code for CLAHE

```
@author: Kalkidan Mulatu
"""
import os
import numpy as np
import cv2 as cv

import os
path =os.chdir(r"D:\LAG_database_part_1\preprocessing\resize 96X96\median
filter\glaucoma median")
i=0
for file in os.listdir():
        img = cv.imread(file, 0)
        clahe = cv.createCLAHE(clipLimit=2.0, tileGridSize=(8,8))
        cl1 = clahe.apply(img)
        cv.imwrite(r"D:\LAG_database_part_1\preprocessing\resize
96X96\median filter\clahe glaucoma
median/"+"clahe_median_glaucoma{}.jpg".format(i), cl1)

        i=i+1
```

End to end CNN

```
label[0:2999]=1
label[3000:6000]=0
data,label=shuffle(inmatrix,label,random_state=2)
```

```
train_data=(data,label)
print(train_data[0].shape)
print(train_data[1].shape)
batch_size=32
numclass=2
(x,y)=(train_data[0],train_data[1])
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.1,
random_state=2)

X_train = X_train.reshape(X_train.shape[0], 1, img_rows, img_cols)
X_test = X_test.reshape(X_test.shape[0], 1, img_rows, img_cols)
X_train = X_train.astype('float64')
X_test = X_test.astype('float64')
# normalizing the data to help with the training
X_train /= 255
X_test /= 255
print('X_train shape:', X_train.shape)
print(X_train.shape[0], 'train samples')
print(X_test.shape[0], 'test samples')
# convert class vectors to binary class matrices
Y_train = np_utils.to_categorical(y_train, numclass)
Y_test = np_utils.to_categorical(y_test, numclass)

i = 11
start = timeit.default_timer()

model = Sequential()
model.add(Conv2D(16,(3, 3),padding='same',input_shape=(1,96, 96)))
model.add(MaxPooling2D(pool_size=(5,5),padding='same'))
convout1 = Activation('relu')
model.add(convout1)

model.add(Conv2D(32,(3, 3),padding='same'))
model.add(MaxPooling2D(pool_size=(3,3),padding='same'))
convout2 = Activation('relu')
model.add(convout2)

model.add(Conv2D(64, (5, 5),padding='same'))
model.add(MaxPooling2D(pool_size=(3,3),padding='same'))
convout3 = Activation('relu')
model.add(convout3)

model.add(Conv2D(128, (3, 3),padding='same'))
model.add(MaxPooling2D(pool_size=(3,3),padding='same'))
convout4 = Activation('relu')
model.add(convout4)

model.add(Conv2D(256, (5, 5),padding='same'))
model.add(MaxPooling2D(pool_size=(3,3),padding='same'))
convout5 = Activation('relu')
model.add(convout5)

model.add(Conv2D(512, (3, 3),padding='same'))
model.add(MaxPooling2D(pool_size=(5,5),padding='same'))
convout6 = Activation('relu')
model.add(convout6)
model.add(Flatten())
model.add(Dense(512))
model.add(Activation('relu'))
model.add(Dropout(0.2))
model.add(Dense(numclass))
model.add(Activation('softmax'))

print(model.summary())
optimiser = keras.optimizers.Adam(learning_rate=0.001)
model.compile(loss='categorical_crossentropy', optimizer=optimiser
,metrics=['accuracy'])
hist=model.fit(X_train, Y_train, batch_size=batch_size, epochs=100,
verbose=1, validation_split=0.1)
```

Confusion matrix

```
def plot_confusion_matrix(cm, target_names, title='Confusion matrix',
cmap=plt.cm.Blues, normalize=True ):



    import itertools
    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]

        plt.imshow(cm, interpolation='nearest', cmap=cmap,origin='upper')
    plt.title(title)

    tick_marks = np.arange(len(target_names))
    plt.xticks(tick_marks, target_names, rotation=45)
    plt.yticks(tick_marks, target_names)
    plt.grid(False)

    fmt = '.4f'
    thresh = cm.max() / 5.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, format(cm[i, j], fmt),
                 horizontalalignment="center",
                 color="white" if cm[i, j] > thresh else "blue")
    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')
    plt.grid(False)
    plt.show()

target_names = ['glaucoma','healthy']
target_names=[1,0]
cnf_matrix = confusion_matrix(y_test, y_pred)
np.set_printoptions(precision=2)
```

```
print("Confusion Matrix :\n"+str(cnf_matrix))
plot_confusion_matrix(cnf_matrix, target_names)


# Plot non-normalized confusion matrix
plt.figure()
plot_confusion_matrix(cnf_matrix,
                      target_names=['glaucoma','healthy'])

train_loss=hist.history['loss']
val_loss=hist.history['val_loss']
train_acc=hist.history['accuracy']
val_acc=hist.history['val_accuracy']
xc=range(100)
plt.figure(1,figsize=(7,5))
plt.plot(xc,train_loss)
plt.plot(xc,val_loss)
plt.xlabel('num of Epochs')
plt.ylabel('loss')
plt.title('train_loss vs valiation_loss')
plt.grid(True)
plt.legend(['training','validation'])
print(plt.style.available) # use bmh, classic,ggplot for big pictures
plt.style.use(['classic'])
plt.figure(2,figsize=(7,5))
plt.plot(xc,train_acc)
plt.plot(xc,val_acc)
plt.xlabel('num of Epochs')
plt.ylabel('accuracy')
plt.title('training_acc vs validation_acc')
plt.grid(True)
plt.legend(['training','validation'],loc=1)
#print plt.style.available # use bmh, classic,ggplot for big pictures
plt.style.use(['classic'])
```