TYPE CLASSIFICATION OF DIABETIC RETINOPATHY AND GLAUCOMA DETECTION USING DEEP LEARNING

A PROJECT REPORT

Submitted By

BRINDHA P. 312215104019

SABARISH S. 312215104084

VIJAI SAI B. 312215104314

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

SSN COLLEGE OF ENGINEERING

KALAVAKKAM 603110

ANNA UNIVERSITY :: CHENNAI - 600025

April 2019

ANNA UNIVERSITY : CHENNAI 600025

BONAFIDE CERTIFICATE

Certified that this project report titled **"TYPE CLASSIFICATION OF DIABETIC RETINOPATHY AND GLAUCOMA DETECTION USING DEEP LEARNING"** is the *bonafide* work of "**BRINDHA P. (312215104019)**, **SABARISH S. (312215104084)**, and **VIJAI SAI B. (312215104314)**" who carried out the project work under my supervision.

DR. CHITRA BABU
HEAD OF THE DEPARTMENT
Professor,
Department of CSE,
SSN College of Engineering,
Kalavakkam - 603 110

DR. G. RAGHURAMAN
SUPERVISOR
Associate Professor,
Department of CSE,
SSN College of Engineering,
Kalavakkam - 603 110

Place:
Date:

Submitted for the examination held on. . . . . .

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENTS

We thank GOD, the almighty for giving me strength and knowledge to do this project.

We would like to thank and deep sense of gratitude to our guide **Dr. G. RAGHURAMAN**, Associate Professor, Department of Computer Science and Engineering, for his valuable advice and suggestions as well as his continued guidance, patience and support that helped us to shape and refine our work.

Our sincere thanks to **Dr. CHITRA BABU**, Professor and Head of the Department of Computer Science and Engineering, for her words of advice and encouragement and we would like to thank our project Coordinator **Dr. S. SHEERAZUDDIN**, Associate Professor, Department of Computer Science and Engineering for his valuable suggestions throughout the project.

We express our deep respect to the founder **Dr. SHIV NADAR**, Chairman, SSN Institutions. We also express our appreciation to our **Dr. S. SALIVAHANAN**, Principal, for all the help he has rendered during this course of study.

We would like to extend our sincere thanks to all the teaching and non-teaching staffs of our department who have contributed directly and indirectly during the course of our project work. Finally, we would like to thank our parents and friends for their patience, cooperation and moral support throughout our life.

BRINDHA P. **SABARISH S.** **VIJAI SAI B.**

ABSTRACT

Diabetic retinopathy (DR) and Glaucoma are the leading cause of blindness in the working age population all over the world. The diagnosis of DR and Glaucoma through colour fundus images requires experienced clinicians to identify the presence and significance of many small features which, along with complex grading system, makes this difficult and time consuming task. Here we propose a CNN approach to diagnosing DR from fundus images and accurately classifying its severity. We develop a network with Convolutional Neural Network (CNN) architecture and data augmentation which can identify the intricate features involved in the classification task such as micro aneurysms, exudate and haemorrhages on the retina. We train this network using a high-end graphics processor unit (GPU). An open source Kaggle dataset is used as an input for DR and RIGA dataset is used as an input for Glaucoma. Total number of 25000 images are used for diabetic retinopathy and the testing accuracy for DR is 86%. Total number of 2664 images are used in glaucoma and the testing accuracy for glaucoma is 94%.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

There are several eye diseases in the world. The two major eye diseases which most of the people currently have are Diabetic Retinopathy (DR) and Glaucoma. Diabetic retinopathy affects blood vessels in the light-sensitive tissue called the retina that lines the back of the eye. It is the most common cause of vision loss among people with diabetes. High blood sugar from diabetes is associated with damage to the tiny blood vessels in the retina, leading to diabetic retinopathy. Diabetic retinopathy can cause blood vessels in the retina to leak fluid or hemorrhage (bleed), distorting vision. In its most advanced stage, new abnormal blood vessels proliferate (increase in number) on the surface of the retina, which can lead to scarring and cell loss in the retina. Diabetic retinopathy progresses through four stages:

1. Mild Nonproliferative Retinopathy
2. Moderate Nonproliferative Retinopathy
3. Severe Nonproliferative Retinopathy
4. Proliferative Diabetic Retinopathy (PDR)

Mild Nonproliferative Retinopathy: Small areas of balloon-like swelling in the tiny blood vessels of the retina, called microaneurysms, occur at this earliest stage of the disease. These microaneurysms may leak fluid into the retina.

Moderate Nonproliferative Retinopathy: As the disease progresses, blood

vessels that nourish the retina may swell and distort. They may also lose their ability to transport blood. Both conditions cause characteristic changes to the appearance of the retina.

Severe Nonproliferative Retinopathy: Many more blood vessels are blocked, depriving blood supply to areas of the retina. These areas secrete growth factors that signal the retina to grow new blood vessels.

Proliferative Diabetic Retinopathy: At this advanced stage, growth factors secreted by the retina trigger the proliferation of new blood vessels, which grow along the inside surface of the retina and into the vitreous gel, the fluid that fills the eye. The new blood vessels are fragile, which makes them more likely to leak and bleed. This leads to permanent vision loss.

The next major disease is Glaucoma. Glaucoma is a condition that causes damage to your eye's optic nerve and gets worse over time. It's often linked to a buildup of pressure inside your eye. Glaucoma tends to be inherited and may not show up until later in life. The increased pressure, called intraocular pressure, can damage the optic nerve, which transmits images to your brain. If the damage continues, glaucoma can lead to permanent vision loss. Without treatment, glaucoma can cause total permanent blindness within a few years. There are two types of Glaucoma

1. Open Angle Glaucoma
2. Angle Closure Glaucoma

Open Angle Glaucoma: This is the most common type. In open angle glaucoma, aqueous fluid drains too slowly and pressure inside the eye builds up. It usually results from aging of the drainage channel, which doesn't work as well over time.

Angle Closure Glaucoma: Those of Asian and Native American descent are at higher risk for this form of glaucoma. It occurs when the drainage system of the eye becomes blocked. It causes a sudden rise in pressure, requiring immediate, emergency medical care. The signs are usually serious and may include blurred vision, severe headaches, eye pain, nausea, vomiting or seeing rainbow-like halos around lights

1.1 Problem Statement

The objective of the project is to design a convolutional neural network for classification of diabetic retinopathy and detection of glaucoma using deep learning. Images from an open source kaggle dataset is used for diabetic retinopathy. Diabetic retinopathy occurs due to increase in blood glucose level. The dataset contains images of 5 classes as follows:

- class 0 for Normal
- class 1 for Aneurysm(Mild)
- class 2 for Hemorrhage(Moderate)
- class 3 for Hard exudate(Severe)
- class 4 for Proliferative Diabetic Retinopathy(Very Severe)

An open source RIGA dataset is used for glaucoma. Glaucoma is a condition of eye in which optic nerve gets damaged due to pressure referred as intraocular

pressure built up inside the eye. In the existing work of diabetic retinopathy, they have extracted features to find out the diabetis. In our work, we aim to design a Convolutional Neural Network with the Convolution layer, Max Pooling layer, Rectified Linear Unit (ReLU) layer and Fully Connected layer. The first part of the project aims to classify the diabetic retinopathy and rest of our project deals with detection of glaucoma. Most of the work in the field of diabetic retinopathy has been based on disease detection or manual extraction of features, but this paper aims at diagnosis of the disease into its different stages using deep learning. So that we can improve the efficiency.

1.2 Motivation

Diabetic retinopathy occurs due to increase in blood glucose level. There are many different conditions that can affect the eye. Some are minor and resolve by themselves quickly, but others could be serious and lead to serious visual impairment. The vision loss or impairment can be sudden or gradual and some serious eye problems can lead to total loss of vision. The decrease in vision is usually one of the first presenting symptoms and should always be investigated. Blindness can be the result of serious untreated eye problems. DR is a diabetes complication that affects eyes. It is caused by damage to blood vessels of the light sensitive tissue at back of the eye (retina).

Glaucoma is a condition of eye in which optic nerve gets damaged due to pressure referred as intraocular pressure built up inside the eye. Whenever the fluid is not circulating normally, it will accumulate in front of the eye and exerts pressure. Due to the increased pressure, it damage the optic nerve. If the damage

continues, Glaucoma can lead to permanent vision loss. Fundus image plays a major role in diagnosis of Diabetic Retinopathy and Glaucoma.

1.3 Classification Techniques

There are several methods and techniques for image classification. The image classification techniques are classified into Automatic, Manual and Hybrid techniques. The Automatic classification techniques are further classified into Supervised and Unsupervised classification techniques. The different methods of the above-mentioned classification techniques are discussed in the following sections.

1.3.1 Automatic Classification

Automatic image classification methods use algorithms that are applied systematically to the entire image to group pixels into meaningful categories. The majority of the classification methods fall under this category. Automatic image classification methods are further classified into two categories:
1) Supervised classification method
2) Unsupervised classification method

1.3.1.1 Supervised Classification method

Supervised classification methods require input from an analyst. The input from an analyst is known as the training set. Training sample is the most important

factor in the supervised image classification methods. The accuracy of the methods highly depends on the samples taken for training. Training samples are two types, one used for classification and another for supervised classification accuracy. The training set is provided before classification is executed.

1.3.1.2 Unsupervised Classification method

Unsupervised classification is where the outcomes are based on the software analysis of an image without the user providing sample classes. The computer uses techniques to determine which pixels are related and groups them into classes. However, the user must have knowledge of the area being classified when the groupings of pixels with common characteristics produced by the computer have to be related to actual features.

1.3.2 Manual Classification

Manual image classification methods are robust, effective and efficient methods. But manual methods consume more time. In manual methods, the analyst must be familiar with the area covered by the image. Efficiency and accuracy of the classification depends on analyst knowledge and familiarity towards the field of study.

1.3.3 Hybrid Classification

Hybrid image classification methods combine the advantages of automated and manual methods. The hybrid approach uses automated image classification methods to do initial classification, further manual methods are used to refine classification and correct errors.

1.4 Convolution Neural Networks

Convolution Neural Network (CNN) is a system of interconnected artificial neurons that have learnable weights and biases. Convolution Neural Network use relatively little preprocessing compared to other image classification algorithms. The connections have numeric weights that are tuned during the training process so that a properly trained network will respond correctly when presented with an image or pattern to recognize.
The architecture and working of the CNN have been explained in detail in Chapter 3.

1.5 Outline of the Report

The report is divided into 5 chapters. The introduction to the problem is defined chapter 1. The chapter 1 also discusses the different image classification techniques and its advantages and disadvantages. The introduction to Convolution Neural Network is also given in this chapter.
Chapter 2 presents the literature survey undertaken. This chapter discusses in

detail about the papers referred for the better understanding of the problem and explains the techniques carried out in each paper and how it has been applied to this project.

Chapter 3 introduces the concept and architecture of a Convolution Neural Network and their functions and operations.

Chapter 4 deals with outline of the system design and explains how it is carried out for both the diseases.

Chapter 5 deals with the implementation of a Convolution Neural Network for the classification of the images into different classes.

Chapter 6 presents a brief conclusion of this report and the future scope of this project.

CHAPTER 2

LITERATURE SURVEY

2.1 Analysis of Retinal Blood Vessels Using Image Processing Techniques

Dr. M. Renuka Devi, attempts the earlier detection of eye diseases such as glaucoma and diabetic retinopathy [14]. In this paper, they present a novel automated hybrid approach for blood vessel extraction using mathematical morphology and a fuzzy clustering algorithm. The method involves two main steps. In the first step, gray mathematical morphology theories are given to smooth and strengthen the retinal images in order to remove the background and enhance the brightness of retinal blood vessels. In the second step, a fuzzy clustering algorithm is employed to extract retinal blood vessels followed by a purification procedure.

Automatic methods of exudates detection on low-contrast images taken from non-dilated pupils, has two main segmentation steps which are coarse segmentation using Fuzzy C-Means clustering and fine segmentation using morphological reconstruction. Four features, namely intensity, standard deviation on intensity, hue and adapted edge, were selected for coarse segmentation.

Several image processing methods and filters are in practise to detect and extract the attributes of retinal blood vessels such as length, width, pattern and angles. Automated Digital image processing techniques and methods has to undergo more of improvisation to achieve precise accuracy to study the condition of

Retinal Vessels especially in cases of Glaucoma and diabetic retinopathy. They have explained various Templates based matched filters, thresholding Methods, segmentation methods, and functional approaches to isolate the blood vessels.

2.2 Detection of Diabetic Retinopathy in Blurred Digital Fundus Images

Eman M. Shahin, attempts the main stages of diabetic retinopathy are Non-Proliferative Diabetic Retinopathy (NPDR) and Proliferative Diabetic Retinopathy (PDR). In this paper, they propose a system for automated classification of normal, and abnormal retinal images by automatically detecting the blood vessels, hard exudates microaneurysms, entropy and homogeneity. The objective measurements such as blood vessels area, exudates area, microaneurysms area, entropy and homogeneity are computed from the processed retinal images. These objective measurements are finally fed to the Artificial Neural Network (ANN) classifier for the automatic classification. Different approaches for image restoration are tested and compared on Fundus images. The effect of restoration on the automatic detection process is investigated in this paper [6].

Exudates appear as bright yellow-white deposits on the retina due to the leakage of blood from abnormal vessels. Their shape and size will vary with the different retinopathy stages. The Fundus image is first preprocessed to standardize its size to 576x720. In order to detect exudates, firstly similar to blood vessels detection, green component of the RGB image is extracted. Two structuring elements (SEs), namely octagon shaped SEs and disc-shaped SEs, were used. A morphological

closing operation was performed using an octagon-shaped SE of size 9. This results in a good contrast image between the exudates and the background. However, the optic disc will also be present together with the exudates, as their gray levels are comparable with those of exudates. Column wise neighborhood operation was performed to remove the unwanted background artifacts, leaving only the exudates and the optical disc. Exudates have irregular shapes and borders. To solve this problem, thresholding is performed to the image with the threshold value of 0.7. Then morphological closing with a disc shaped SE of size 10 is used to fill up the holes or gaps of the exudates. The optic disc contains the highest pixel values in the image. Therefore, to remove the optic disc, edge detection using Canny method is used together with a Region of Interest (ROI). First, a radius of 82 is defined as most optic discs are of size 80 x 80 pixels. Next, the optic disc is removed together with the border. Finally, by performing a morphological erosion operation with disk shaped structuring element (SE) of size 3, exudates were extracted.

2.3 Diagnosis of Diabetic Retinopathy Using Morphological Process and SVM Classifier

Mahendran Gandhi, attempts focuses on automatic detection of diabetic retinopathy through detecting exudates in colour fundus retinal images and also classifies the rigorousness of the lesions. Decision making of the severity level of the disease was performed by SVM classifier [10].

In biomedical applications, automated retinal image analysis made the detection of retinal pathologies much easier for ophthalmologists, whereas conventional

methods, such as dilating the eye pupil, take time and make patients suffer for a while. Diabetic Retinopathy is caused by the damage to blood vessels of the retina. It occurs when high blood glucose, the characteristic of diabetes, has damaged the small vessels that provide oxygen and nutrients to the retina.This paper also focuses on exudates for the reason that it provides information about earlier diabetic retinopathy. The proteins and lipids getting leaked from the bloodstream into the retina through damaged blood vessels is the chief cause of exudates.

The screening process for diabetic retinopathy involves excessive dilation of pupil which affects the patients eye. So, an automated method is presented in this paper for detection of exudates from the non-dilated colour fundus retinal images using morphological process. Edge detection algorithm is applied on the preprocessed image to make it suitable for detecting the optic disc, blood vessels and exudates. Canny edge detector is employed for the contour detection. This algorithm finds the edges where the gray scale intensity of the image changes and this variation can be found by determining gradients of the image. It enhances the blurred edges by preserving all local maxima in the gradient image. It can detect the boundaries optimally. Then, the mask image was created in which region of interest in the image is optic disc. The region outside the optic disc is appeared to be dark. The masked image is subtracted from the edge detected image in order to eliminate the optic disc.

The input features required for the SVM classifier were extracted using Gray Level Co-occurrence Matrix (GLCM). It contains information about the positions of pixels having similar gray level values. It can make use of distance vector.The gray-level co-occurrence matrix is represented as G[i,j] which is used to calculate all pair of pixels separated by distance vector having gray levels i and j. Based on

the analyzed matrix and the texture information, the following features like entropy, contrast, correlation, energy, homogeneity and dissimilarity were obtained.

This classifier is used to evaluate training data to find a best way to classify images into different cases like moderate or severe. To classify the disease, the relevant features of an image should be extracted.

2.4 Optic Disc and Cup Segmentation Methods for Glaucoma Detection with Modification of U-Net Convolutional Neural Network

Glaucoma is the second leading cause of blindness all over the world, with approximately 60 million cases reported worldwide in 2010, and an increase by 20 million is expected in 2020. If left unnoticed, glaucoma can cause irreversible damage to the optic nerve leading to blindness. Therefore, diagnosing glaucoma at early stages is very important [2]. Glaucoma is the second leading cause of blindness all over the world, with approximately 60 million cases reported worldwide in 2010. If undiagnosed in time, glaucoma causes irreversible damage to the optic nerve leading to blindness.

The optic nerve head examination, which involves measurement of cup-to-disc ratio, is considered one of the most valuable methods of structural diagnosis of the disease. Estimation of cup-to-disc ratio requires segmentation of optic disc and optic cup on eye fundus images and can be performed by modern computer vision algorithms. This work presents universal approach for automatic optic disc

and cup segmentation, which is based on deep learning, namely, modification of U-Net convolutional neural network. For both optic disc and cup segmentation, this method achieves quality comparable to current state-of-the-art methods, outperforming them in terms of the prediction time.

2.5 Comparison of Image Preprocessing Techniques on Fundus Images for Early Diagnosis of Glaucoma

In this paper, S.Rathinam proposed a comparisoin of the performance of five preprocessing techniques namely Contrast adjustment, Adaptive Histogram equalization,Median filtering, Average filtering and Homomorphic filtering. The performance of these techniques are evaluated using Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR)[13]. Preprocessing of eye fundus image is a crucial initial step before further analysis is performed. Many preprocessing techniques are available.

Contrast adjustment: The contrast of an image is the distribution of its dark and light pixels. A low-contrast image exhibits small differences between its light and dark pixel values. The histogram of a low-contrast image is narrow. Since the human eye is sensitive to contrast rather than absolute pixel intensities, a perceptually better image could be obtained by stretching the histogram of an image so that the full dynamic range of the image is filled.

Adaptive Histotram Equalization: Adaptive histogram equalization (AHE) is a computer image processing technique used to improve contrast in images. It

differs from ordinary histogram equalization in the respect that the adaptive method computes several histograms, each corresponding to a distinct section of the image, and uses them to redistribute the lightness values of the image.

Average filtering: The Average (mean) filter smooths image data, thus eliminating noise. This filter performs spatial filtering on each individual pixel in an image using the grey level values in a square or rectangular window surrounding each pixel. The average filter computes the sum of all pixels in the filter window and then divides the sum by the number of pixels in the filter window.

Median filtering: The Median Filter does somewhat the same, but instead of taking the mean or average, it takes the median. The median is gotten by sorting all the values from low to high, and then taking the value in the center. If there are two values in the center, the average of these two is taken. A median filter gives better results to remove salt and pepper noise, because it completely eliminates the the noise.

Homomorphic filtering: Homomorphic filter is sometimes used for image enhancement. It simultaneously normalizes the brightness across an image and increases contrast. Here homomorphic filtering is used to remove multiplicative noise.

These algorithms are evaluated using Peak Signal to Noise Ratio and Mean Square Error. The Median filtering and Average filtering give suitable results and Median filter is found to be better with high PSNR and low MSE values.

CHAPTER 3

CONVOLUTIONAL NEURAL NETWORKS

3.1 Introduction

A system of interconnected artificial neurons that have learnable weights and biases forms a Convolutional Neural Network. These neurons exchange messages between each other. The connections have numeric weights that are tuned during the training process, so that a properly trained network will respond correctly when presented with an image or pattern to recognize. The network consists of multiple layers of feature-detecting neurons. Each layer has many neurons that respond to different combinations of inputs from the previous layers. As shown in Figure 3.1, the layers are built up so that the first layer detects a set of primitive patterns in the input, the second layer detects patterns of patterns, the third layer detects patterns of those patterns.

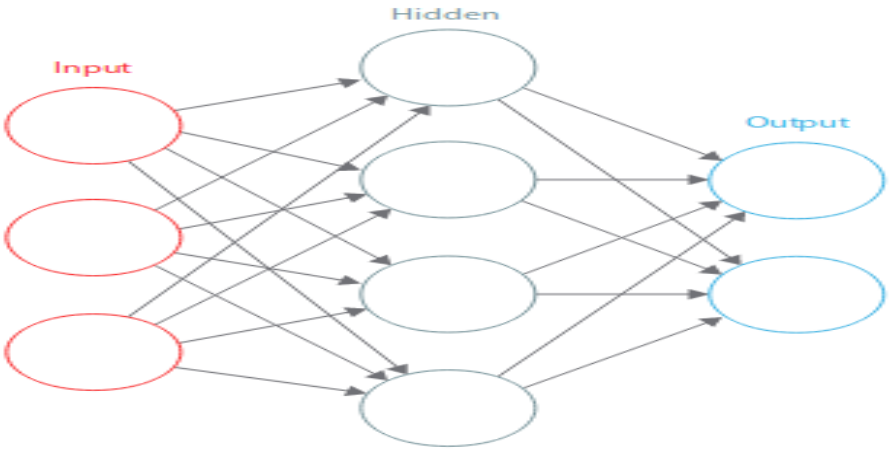

FIGURE 3.1: An Artificial Neural Network

3.2 Layers of CNN

For classification problems, complex architectures are built by stacking multiple and different layers in a CNN. The four types of layers are convolution layer, pooling/subsampling layers, non-linear (ReLU) layers, and fully connected layers. Figure 3.2 shows the various layers of CNN. A portion of the input image is fed to the convolution layer. The output of this layer is then fed to the pooling layer. This is repeated again followed by fully connected layer which performs classification.

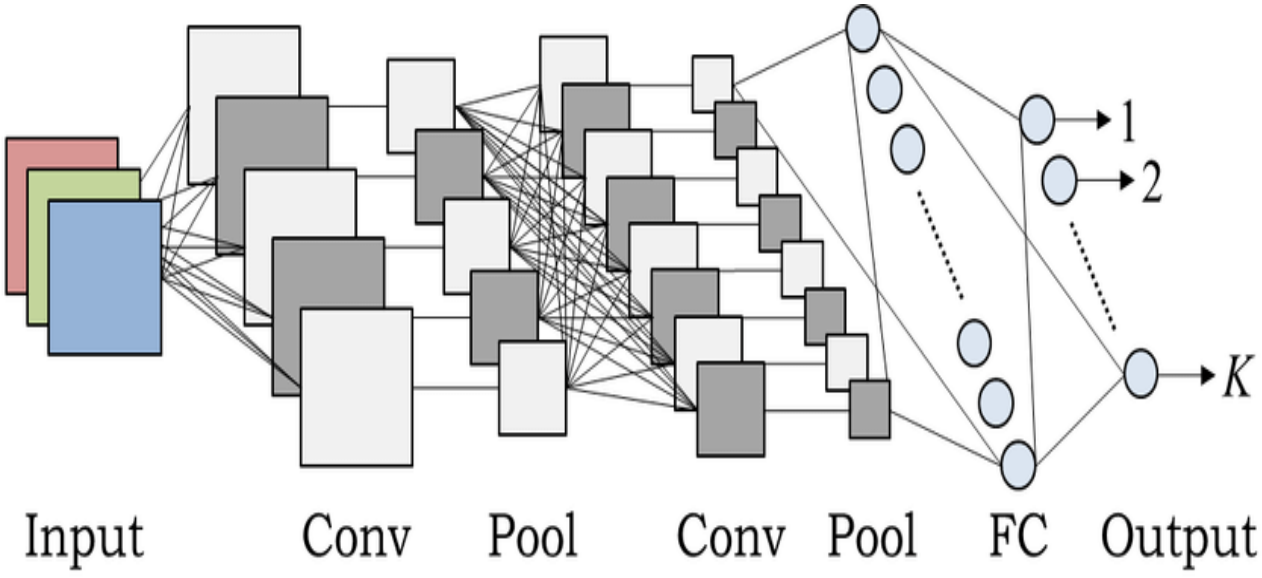

FIGURE 3.2: Architecture of CNN

3.2.1 Convolutional Layer

Different features of the input are extracted by the convolution operation. LThe first convolution layer extracts low-level features like edges, lines, and corners.

Higher-level layers extract higher-level features. Figure 3.3 illustrates the process of 3D convolution used in CNNs. The input is of size N x N x D and is convolved with H kernels, each of size k x k x D separately. Convolution of an input with one kernel produces one output feature, and with H kernels independently produces H features. Starting from top-left corner of the input, each kernel is moved from left to right, one element at a time. Once the top-right corner is reached, the kernel is moved one element in a downward direction, and again the kernel is moved from left to right, one element at a time. This process is repeated until the kernel reaches the bottom-right corner.

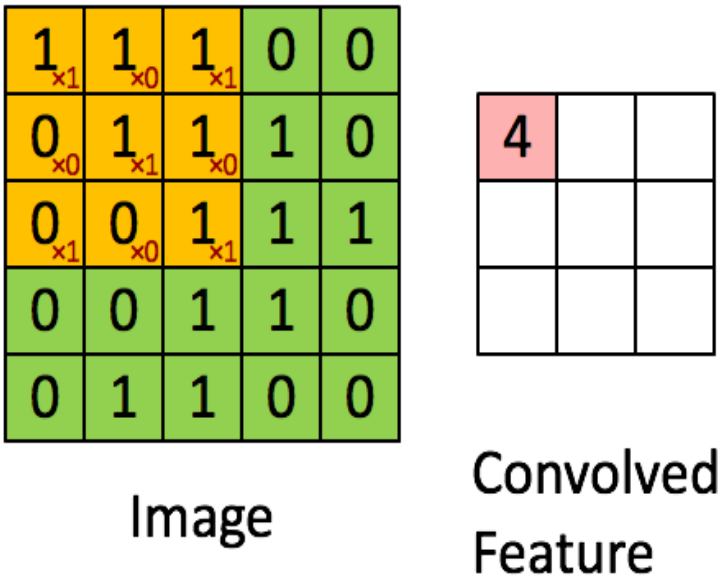

FIGURE 3.3: Representation of Convolutional Process

For example, if N = 5 and k = 5 , there are 5 separate positions from left to right and 5 separate positions from top to bottom that the kernel can take. Corresponding to these positions, each feature in the output will contain 28x28 (i.e., (N-k+1) x (N-k+1)) elements. For each position of the kernel in a sliding window process, k x k x D elements of input and k x k x D elements of kernal are element-by-element

multiplied and accumulated. So to create one element of one output feature, k x k x D multiply-accumulate operations are required.

3.2.2 Pooling Layer

The pooling (subsampling) layer helps to reduce the resolution of the features. The features are robust against noise and distortion. The two ways to perform pooling operation is as follows: max pooling and average pooling. In both the cases, the input is divided into non-overlapping sub-regions.

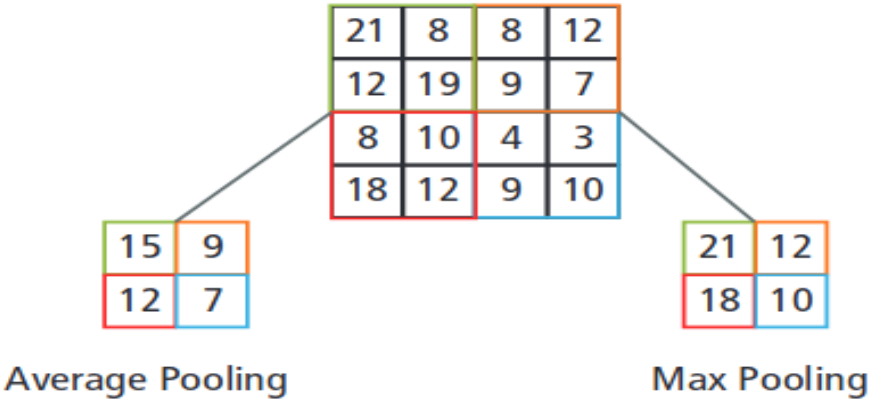

FIGURE 3.4: Representation of Max Pooling and Average Pooling

Figure 3.4 elaborates further on the pooling process. The input is of size 4x4 input image is divided into four non-overlapping matrices each of size 2x2 shown in green, yellow, red and blue squares. In the case of max pooling operation, the maximum value of the four values in the 2x2 matrix (21 from the green matrix, 12 from the yellow matrix, 18 from the red matrix and 10 from the blue matrix) is taken as the output. But in the case of average pooling, the average of the four

values (15 from the green matrix, 9 from the yellow matrix, 12 from the red matrix and 7 from the blue matrix) is taken as the output. If the result of averaging is a fraction, then it has to be rounded to nearest integer.

3.2.3 Non Linear Layer

The ReLU layer implements the function: y=max(x,0). Now the input and output sizes of this layer are the same. This operation increases the nonlinear properties of the decision function including the overall network. This does not affect the respective fields of the convolution layer. Compared to the other non-linear functions used in CNNs (e.g., Sigmoid, hyperbolic tangent and absolute of hyperbolic tangent), the major advantage of a ReLU is that the network trains many times faster. The functionality of ReLU is illustrated in Figure 3.5. The transfer function plotted above the arrow. All the positive values (15, 20, 35, 18, 25, 100, 20, 25, 101, 75, 18, 23) are retained as such and the negative values (-10, -110, -15, -10) are converted to zero.

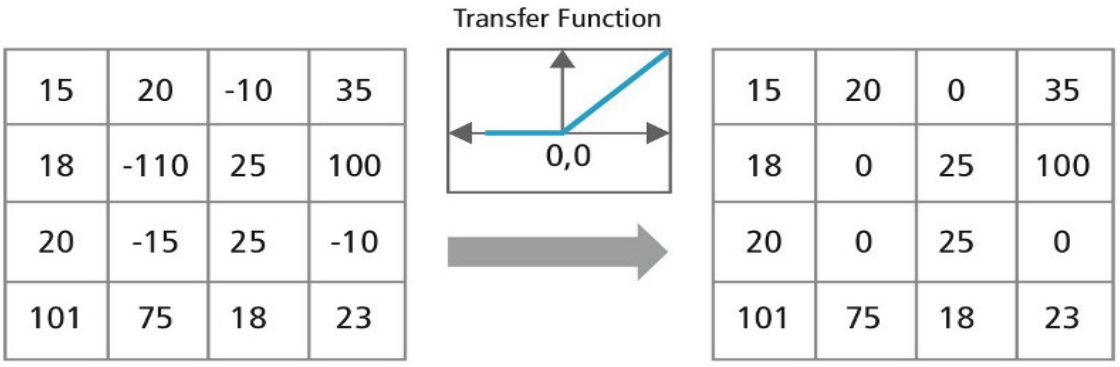

FIGURE 3.5: Representation of ReLU Functionality

3.2.4 Fully Connected Layer

Fully connected layers are generally used as the final layers of a CNN. Mathematically these layers sum a weighting of the previous layer of features. This indicates the accurate mix of ingredients to determine a specific target output result. In case of a fully connected layer, all the elements of all the features of the previous layer are used in the calculation of each element of each output feature. Figure 3.6 explains the fully connected layer L. Layer L-1 has two features, each of which is 2x2. It has four elements. For example, each of the element in the first feature in the (L-1) layer is multiplied with two set of weights and is summed up to give the feature in the (L-1) layer. Layer L has two features, each having a single element.

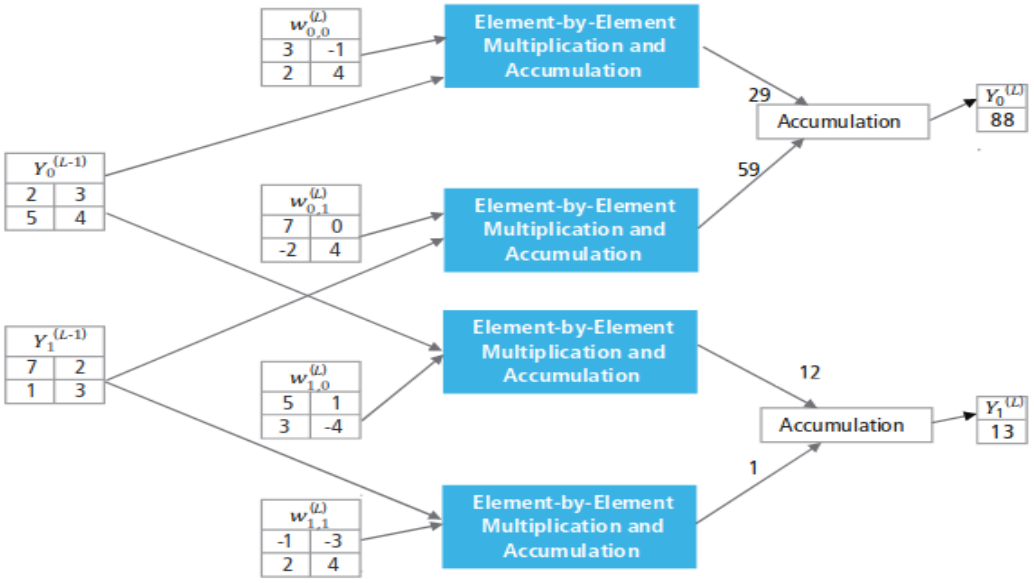

FIGURE 3.6: Processing of Fully Connected Layer

3.2.5 L2 Regularization

Overfitting is a common problem of neural network approaches. Overfitting means that the classification results can be very good on the training data set but poor on the test data set. To avoid overfitting, it is necessary to adopt additional techniques such as regularization. Regularizers allow to apply penalties on layer parameters or layer activity during optimization. These penalties are incorporated in the loss function that the network optimizes. L2 regularization is perhaps the most common form of regularization. It can be implemented by penalizing the squared magnitude of all parameters directly in the objective. That is, for every weight w in the network, we add the term $\frac{1}{2}\lambda w^2$ to the objective, where λ is the regularization strength. It is common to see the factor of $\frac{1}{2}$ in front because then the gradient of this term with respect to the parameter w is simply λw instead of $2\lambda w$. The L2 regularization has the intuitive interpretation of heavily penalizing peaky weight vectors and preferring diffuse weight vectors.

3.3 Advantages of CNN

Neural networks and other pattern detection methods have been around for the past 50 yeras. In the recent past, there has been a significiant development in the area of convolutional neural networks.

Ruggedness to shifts and distortion in the image:

Distortions like a change in shape due to the camera lens, different lighting conditions, different poses, presence of partial occlusions, horizontal and vertical

shifts, etc. does not affect the CNN during detection. Since the same weight configuration is used across space. CNNs are shift invariants. Using fully connected layers we can also achieve shift invariants. In this case multiple units with identical weight patterns at different locations of the input is the outcome of the training. A large number of training instances would be required to learn these weight configurations so as to cover the space of possible variations.

Fewer memory requirements:

When a fully connected layer is used to extract the features, if the input image of size 32x32 and a hidden layer having 1000 features, it will require an order of 10^6 coefficients. This is a huge memory requirement. But when we use a convolutional layer, the same coefficients are used across different locations in the space. Hence the memory requirement is drastically reduced.

Easier and better training:

The standard neural network is equivalent to a CNN, will have number of parameters. The training time would also increase proportionally. Since the number of parameters is drastically reduced in CNN, training time is also proportionately reduced. A standard neural network can be designed whose performance is as same as CNN with an assumption of perfect training. In practical training, a standard neural network equivalent to CNN would have more parameters. This leads to more noise addition during the training process. Therefore, the performance of a standard neural network equivalent to a CNN will always be poorer.

Better Accuracy:

The accuracy of a convolutional neural network is superior in image recognition problems. CNNs are found to have better accuracy mean compared to other classification techniques. Table 3.1 from the paper by Joseph Lemley et al, describes the mean and standard deviation for various classification techniques. It is understood from that the mean and the standard deviation is more desirable while using CNN's.

TABLE 3.1: Comparison of Mean and Standard Deviation for Various Classification Techniques

S.No	Method	Mean	SD
1.	CNN	97.8%	0.0058
2.	DTCWT on SVM[RBF]	90.7%	0.0047
3.	PCA+SVM[RBF]	90.2%	0.0063
4.	SVM[RBF]	87.1%	0.0053
5.	HOG+SVM[RBF]	85.6%	0.0042
6.	HOG+SVM[linear]	84.6%	0.0024
7.	DTCWT on SVM[linear]	83.3%	0.0047
8.	PCA+SVM[linear]	81%	0.0071
9.	SVM[linear]	76.5%	0.0099

3.4 Applications of CNN

Image Classification: On large scale datasets CNNs acheive better classification accuracy compared to other methods due to their capability of join feature and classifier learning.

Speech Recognition: Convolutional Neural Networks have been used in Speech

Regognition recently. CNN has given better results over Deep Neural Networks (DNN). In 2015, Microsoft Corporation researchers came up with four domains in which CNN give better results than DNN. They are (1) Noise robustness (2) Distant speech recognition (3) Low-footprint models (4) Channel-mismatched training-test conditions.

Face Recognition: Problems like focussing on each face despite bad lighting or different pose, identifying all the faces in the picture, identifying unique features, comparing identified features to existing database and determining the person's name is a part of the face recognition process. Faces represent a complex, multi-dimensional, visual stimulus. It is presented using a hybrid neural network combining local image sampling, a self-organizing map neural network and a convolutional neural network.

Scene Labelling: In scene labelling each pixel is labelled with the category of the object it belongs to, CNNs are very effective in scene labelling.

Action Recognition: The transalations and distortions of features in different patterns which belong to the same action class are the difficulties in developing an action recognition system. These difficulties can be overcome by using CNNs.

Human Pose Estimation: In computer vision, human pose recognition has been a long standing problem. Due to the high variability of possible body poses and high dimensionality of the input data, this problem couldn't be solved. Now CNNs are useful in human pose estimation.

Document Analysis: The sequential nature of the pen trajectory by representing the input in the time domain is used in traditional handwriting recognizers. These representations are sensitive to writing speed, stroke order, and other irrelevant parameters.

3.5 Summary

In this chapter, a detailed description of the Convolutional Neural Network (CNN) was given. The functions of various layers such as convolution, polling, ReLU and fully connected layers was elaborated. We have understood that the convolution layer is used to extract features from the input image, the pooling layer to reduce the dimensionality, the ReLU layer to increase the non-linear properties and the fully connected layer the output of the previous layer that is it takes neurons from the previous layer, connects it to every single neuron it has forming a neural network and then compiling the network for further classification. L2 Regularisation is used to avoid overfitting problem.

CHAPTER 4

SYSTEM DESIGN

4.1 Architecture for DR Classification

The system architecture shown in Figure 4.1 gives the overall view about all the modules in the proposed system of Diabetic Retinopathy and the flow of the process right from data collection to classification. Initially, input images are

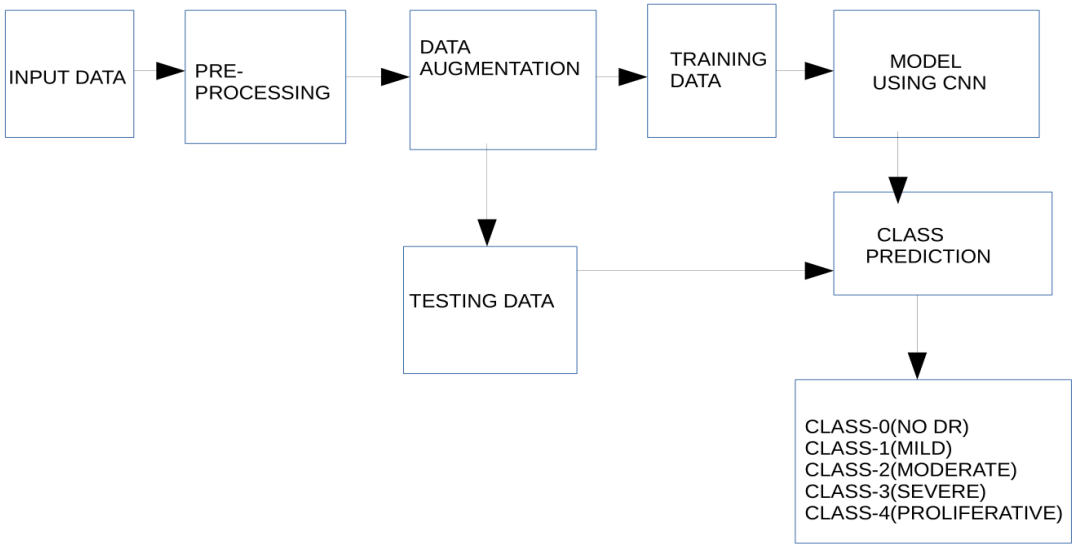

FIGURE 4.1: System Architecture for Diabetic Retinopathy

collected from open-source Kaggle dataset. All the images are of size 512X512. Next, the preprocessing step involves conversion of the RGB to gray scale image. The gray scale images are given as input to the data augmentation. The augmented images are trained using the CNN model. Finally, the class prediction is done with the test data.

4.2 Architecture for Glaucoma Detection

The system architecture shown in Figure 4.2 gives overall view about all the modules in the proposed system of Glaucoma and the flow of the process right from data collection to detection.

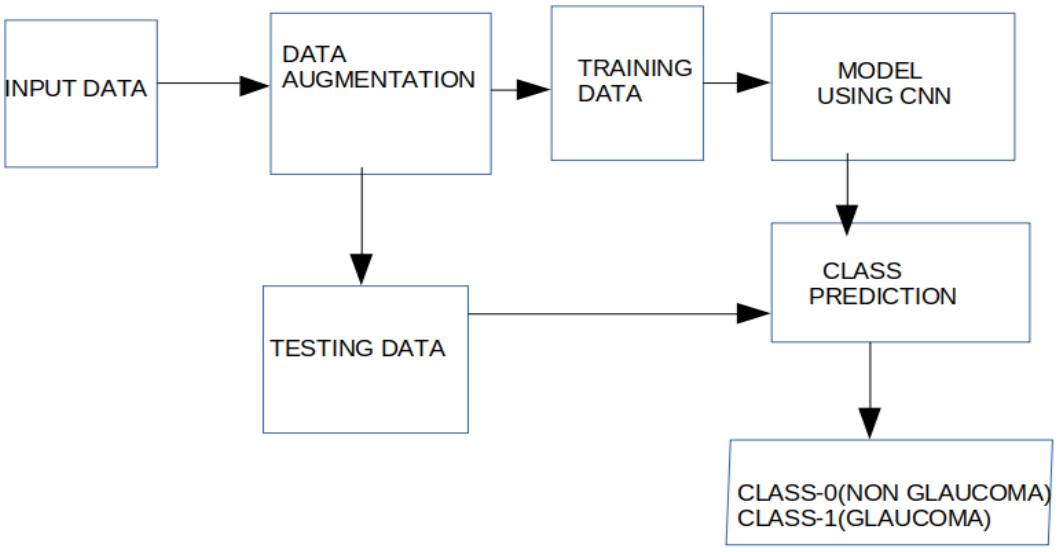

FIGURE 4.2: System Architecture for Glaucoma

Initially, input images are collected from open-source RIGA dataset. All the images are of size 256X256. Next, images are given as input to the data augmentation. The augmented images are trained using the CNN model. Finally, the class prediction is done with the test data.

CHAPTER 5

IMPLEMENTATION

5.1 Image Classification Process

Input: The input consists of a set of images, each labelled with different classes. This is the training set.

Training: The training set is used to learn the details as each of the classes and its position in the image. This step is known as training a classifier or learning a model.

Testing: The quality of the classifier is evaluated by asking it to predict labels for a new set of images that it has never seen before. The more the values are matched the more is the accuracy of the classifier.

5.2 Proposed System-Diabetic Retinopathy

The proposed system describes the step by step process of the DR and gives a detailed view and illustration of the entire process.

5.2.1 Preprocessing

Input images are collected from kaggle open-source dataset. Retinal image is used as the input image. The input retinal images are preprocessed before it is applied to

the further process. In the preprocessing stage, problems arise due to the blurred image or non-clarity images are rectified. This stage involves the colour space conversion, image restoration and enhancement. In photography and computing, a grayscale image is an image in which the value of each pixel is a single sample. Displayed images of this sort are typically composed of shades gray, varying from black at the weakest intensity to white at the strongest, though in principle the samples could be displayed as shades of any color, or even coded with various colors for different intensities. Grayscale images are distinct from black-and-white images, which in the context of computer imaging are images with only two colors, black and white. The grayscale image is shown in Figure 5.1.

FIGURE 5.1: Grayscale Image

5.2.2 Data Augmentation

Data augmentation is a way of creating new data with different orientations. The DR consists of 31,615 images for training which is imbalance in nature. Data Augmentation plays a major role to balance the 5 classes in Diabetic Retinopathy. Data Augmentation like rotation, zooming, shearing, horizontal and vertical flip are done on images of 5 classes. Finally, the class prediction is based on the

weighted values for each training images. Table 5.1 shows the augmented data in each class of DR.

TABLE 5.1: Augmented Dataset for DR

Images	Before Augmentation	After Augmentation
Class 0	23229	5000
Class 1	2199	5000
Class 2	4763	5000
Class 3	786	5000
Class 4	638	5000
Total Images	31615	25000

5.2.3 Architecture of the Network

The first convolutional layer accepts the input image of size 510x510. It consists of 32 filters of size 3x3. The output is then fed into a pooling layer which consists of filters of size 2x2. Since this is the max-pooling operation, the maximum value among the filtered value is chosen. The second convolutional layer consists of 32 filters of size 3x3. The output thus obtained is again fed into a pooling layer which consists of input size 126x126 and filters of size 2x2. The third convolutional layer consists of input size 124x124 and 32 filters of size 3x3. The output is again fed into pooling layer which consists of input size 62x62 and filter of size 2x2. The ReLU layer is a rectified Layer Unit that performs the rectifying operation to rectify the negative pixel values. Finally, the fully connected layer that classifies the pixels into one of the 5 different classes. Figure 5.2 displays each layer of the network described above.

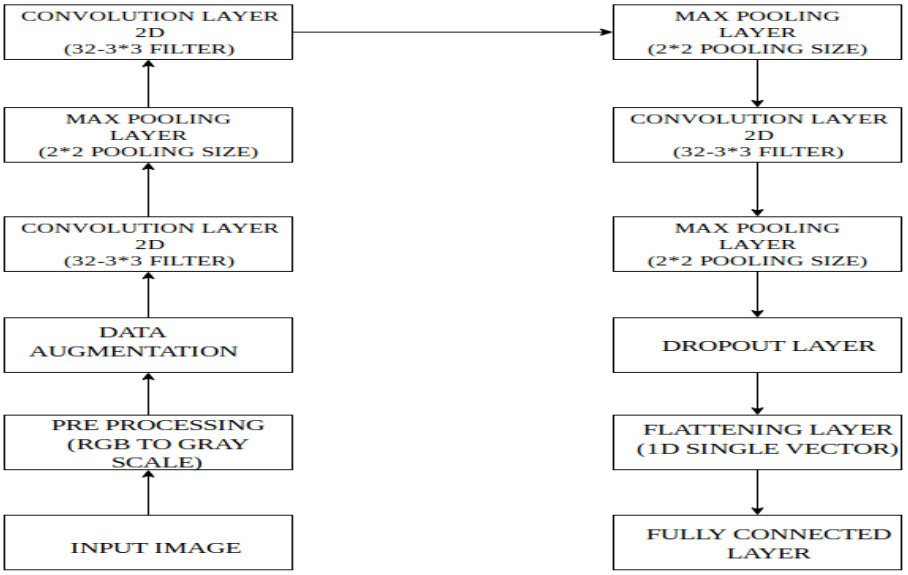

FIGURE 5.2: Network Architecture for DR

5.2.4 Classification Output

A Convolutional Neural Network that can take input images from kaggle dataset and classify it into one of the five classes with multiple convolutional and pooling layers is designed. The model is trained over 32 epochs. The training model is shown in Figure 5.3.

```
Epoch 1/32
781/781 [==============================] - 6521s 8s/step - loss: 1.3024 - acc: 0.3726 - val_loss: 1.8358 - val_acc: 0.2232
Epoch 2/32
781/781 [==============================] - 303s 388ms/step - loss: 1.2479 - acc: 0.3905 - val_loss: 1.9527 - val_acc: 0.2385
Epoch 3/32
781/781 [==============================] - 294s 376ms/step - loss: 1.2297 - acc: 0.4092 - val_loss: 1.9201 - val_acc: 0.2202
Epoch 4/32
781/781 [==============================] - 294s 377ms/step - loss: 1.2051 - acc: 0.4256 - val_loss: 1.8049 - val_acc: 0.2523
Epoch 5/32
781/781 [==============================] - 295s 378ms/step - loss: 1.1439 - acc: 0.4628 - val_loss: 1.6902 - val_acc: 0.2844
Epoch 6/32
781/781 [==============================] - 295s 378ms/step - loss: 1.0319 - acc: 0.5248 - val_loss: 1.4160 - val_acc: 0.3624
Epoch 7/32
781/781 [==============================] - 295s 377ms/step - loss: 0.8827 - acc: 0.6052 - val_loss: 1.3460 - val_acc: 0.3532
Epoch 8/32
781/781 [==============================] - 294s 376ms/step - loss: 0.7342 - acc: 0.6802 - val_loss: 1.0466 - val_acc: 0.4862
Epoch 9/32
781/781 [==============================] - 296s 378ms/step - loss: 0.6186 - acc: 0.7321 - val_loss: 0.9806 - val_acc: 0.5491
Epoch 10/32
781/781 [==============================] - 296s 378ms/step - loss: 0.5178 - acc: 0.7808 - val_loss: 0.7671 - val_acc: 0.7018
Epoch 11/32
781/781 [==============================] - 296s 378ms/step - loss: 0.4624 - acc: 0.8089 - val_loss: 0.6208 - val_acc: 0.7385
Epoch 12/32
781/781 [==============================] - 298s 381ms/step - loss: 0.4154 - acc: 0.8335 - val_loss: 0.5417 - val_acc: 0.7752
Epoch 13/32
781/781 [==============================] - 296s 379ms/step - loss: 0.3710 - acc: 0.8530 - val_loss: 0.5196 - val_acc: 0.7982
Epoch 14/32
781/781 [==============================] - 297s 380ms/step - loss: 0.3387 - acc: 0.8644 - val_loss: 0.4681 - val_acc: 0.7982
Epoch 15/32
781/781 [==============================] - 296s 379ms/step - loss: 0.3146 - acc: 0.8779 - val_loss: 0.4684 - val_acc: 0.7936
Epoch 16/32
781/781 [==============================] - 296s 379ms/step - loss: 0.2990 - acc: 0.8821 - val_loss: 0.3959 - val_acc: 0.8349
Epoch 17/32
781/781 [==============================] - 296s 379ms/step - loss: 0.2843 - acc: 0.8892 - val_loss: 0.4117 - val_acc: 0.8259
Epoch 18/32
781/781 [==============================] - 297s 380ms/step - loss: 0.2682 - acc: 0.8947 - val_loss: 0.4402 - val_acc: 0.8303
Epoch 19/32
781/781 [==============================] - 296s 379ms/step - loss: 0.2587 - acc: 0.9008 - val_loss: 0.3533 - val_acc: 0.8440
Epoch 20/32
781/781 [==============================] - 296s 379ms/step - loss: 0.2422 - acc: 0.9053 - val_loss: 0.2764 - val_acc: 0.8853
Epoch 21/32
781/781 [==============================] - 296s 379ms/step - loss: 0.2266 - acc: 0.9124 - val_loss: 0.4472 - val_acc: 0.8073
Epoch 22/32
781/781 [==============================] - 296s 379ms/step - loss: 0.2310 - acc: 0.9111 - val_loss: 0.4321 - val_acc: 0.8165
Epoch 23/32
781/781 [==============================] - 296s 379ms/step - loss: 0.2232 - acc: 0.9146 - val_loss: 0.3308 - val_acc: 0.8716
Epoch 24/32
781/781 [==============================] - 296s 379ms/step - loss: 0.2149 - acc: 0.9196 - val_loss: 0.3225 - val_acc: 0.8440
Epoch 25/32
781/781 [==============================] - 296s 379ms/step - loss: 0.2022 - acc: 0.9217 - val_loss: 0.2893 - val_acc: 0.8750
Epoch 26/32
781/781 [==============================] - 295s 378ms/step - loss: 0.2054 - acc: 0.9245 - val_loss: 0.2996 - val_acc: 0.8578
Epoch 27/32
781/781 [==============================] - 296s 380ms/step - loss: 0.2009 - acc: 0.9246 - val_loss: 0.2644 - val_acc: 0.8899
Epoch 28/32
781/781 [==============================] - 297s 380ms/step - loss: 0.1937 - acc: 0.9281 - val_loss: 0.2901 - val_acc: 0.8394
Epoch 29/32
781/781 [==============================] - 299s 382ms/step - loss: 0.1860 - acc: 0.9286 - val_loss: 0.3108 - val_acc: 0.8578
Epoch 30/32
781/781 [==============================] - 297s 380ms/step - loss: 0.1848 - acc: 0.9317 - val_loss: 0.2527 - val_acc: 0.8991
Epoch 31/32
781/781 [==============================] - 298s 381ms/step - loss: 0.1793 - acc: 0.9343 - val_loss: 0.3077 - val_acc: 0.8394
Epoch 32/32
781/781 [==============================] - 297s 380ms/step - loss: 0.1767 - acc: 0.9337 - val_loss: 0.2904 - val_acc: 0.8761
```

FIGURE 5.3: Training Model for Diabetic Retinopathy

Among 8256 testing images the overall accuracy is predicted as 86% which is shown in Figure 5.4.

FIGURE 5.4: Overall Accuracy for Diabetic Retinopathy

Figure 5.5 describes sample images tested over class 0. Column 1 describes respective class and column 2 describes the predicted output for all 5 classes which is shown below.

class0	[0]
class0	[0]
class0	[0]
class0	[0]
class0	[0]
class0	[0]
class0	[0]
class0	[0]
class0	[0]
class0	[0]
class0	[0]

FIGURE 5.5: Class 0 (No DR)

Figure 5.6 describes sample images tested over class 1.

class1	[1]
class1	[1]
class1	[1]
class1	[1]
class1	[1]
class1	[2]
class1	[1]
class1	[1]
class1	[1]
class1	[1]
class1	[0]

FIGURE 5.6: Class 1 (Mild)

Figure 5.7 describes sample images tested over class 2.

class2	[2]
class2	[2]
class2	[0]
class2	[1]
class2	[2]
class2	[2]
class2	[2]
class2	[0]
class2	[2]
class2	[2]
class2	[0]

FIGURE 5.7: Class 2 (Moderate)

Figure 5.8 describes sample images tested over class 3.

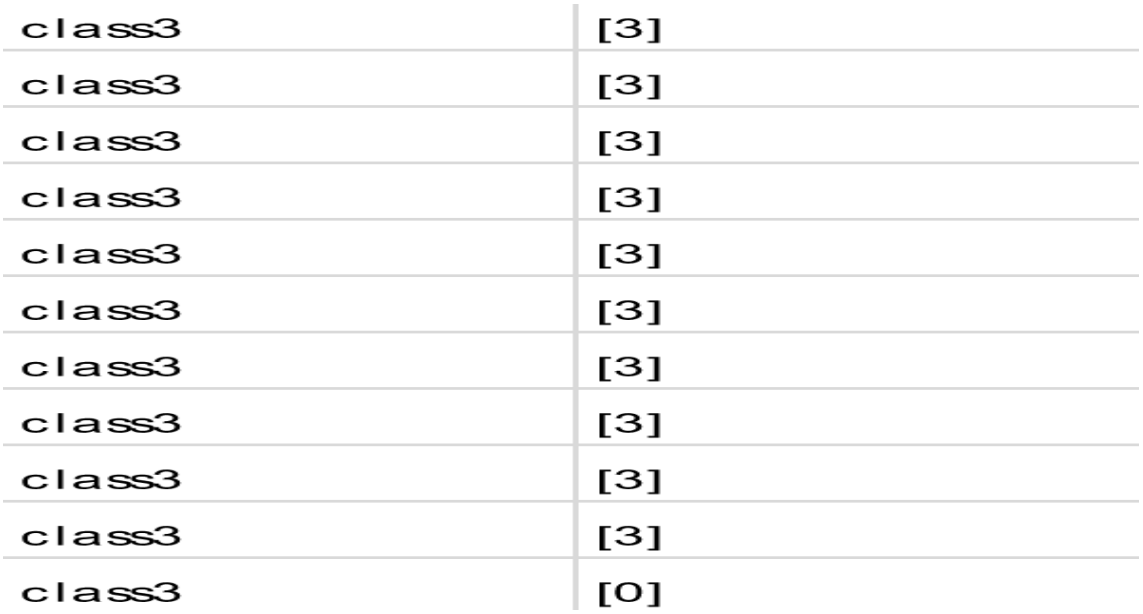

FIGURE 5.8: Class 3 (Severe)

Figure 5.9 describes sample images tested over class 4.

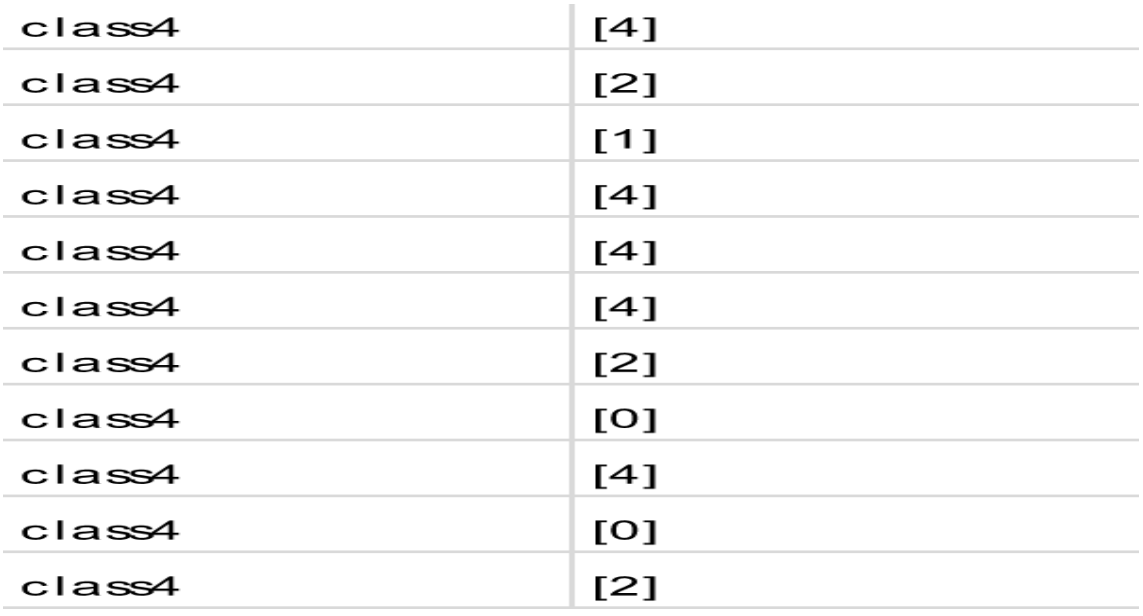

FIGURE 5.9: Class 4 (Proliferative)

The graph for the detailed model for epochs vs accuracy is shown in figure 5.10.

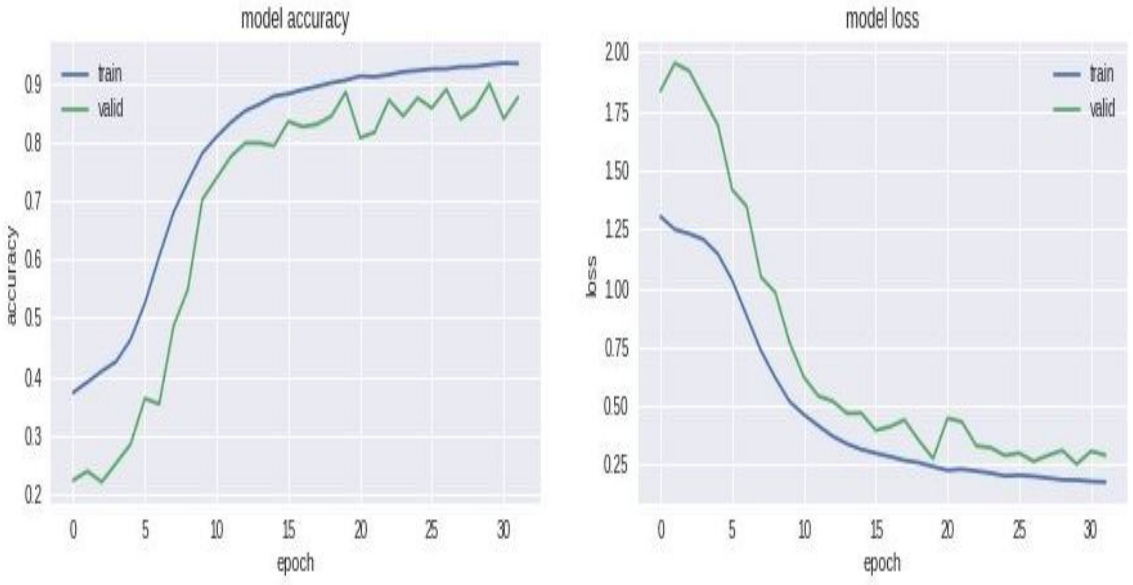

FIGURE 5.10: Graph for the Overall Diabetic Retinopathy

5.3 Proposed System-Glaucoma

The proposed system describes the step by step process of the glaucoma and gives a detailed view and illustration of the entire process.

5.3.1 Data Augmentation

Data augmentation is a way of creating new data with different orientations. Glaucoma consists of 1500 images which is imbalance in nature. Data Augmentation plays a major role to balance the 2 classes in Glaucoma. Data Augmentation like horizontal flip are done on images to balance the classes.

Finally, the class prediction is based on the weighted values for each training images.Table 5.2 shows the augmented data in each class of Glaucoma.

TABLE 5.2: Augmented Dataset for Glaucoma

Images	Before Augmentation	After Augmentation
Class 0(No Glaucoma)	600	1488
Class 1(Glaucoma)	900	1176
Total Images	31615	25000

5.3.2 Architecture of the Network

The first convolutional layer accepts the input image of size 256x256. It consists of 32 filters of size 3x3. The output is then fed into a pooling layer which consists of filters of size 2x2. Since this is the max-pooling operation, the maximum value among the filtered value is chosen. The second convolutional layer consists of 32 filters of size 3x3. The output thus obtained is again fed into a pooling layer which consists of filters of size 2x2. The ReLU layer is a rectified Layer Unit that performs the rectifying operation to rectify the negative pixel values. Finally, the fully connected layer that classifies the pixels into one of the 2 different classes. Figure 5.11 displays each layer of the network described above.

5.3.3 Classification Output

A Convolutional Neural Network that can take input images from this dataset and detect it into one of the two classes with two convolutional and pooling layers is

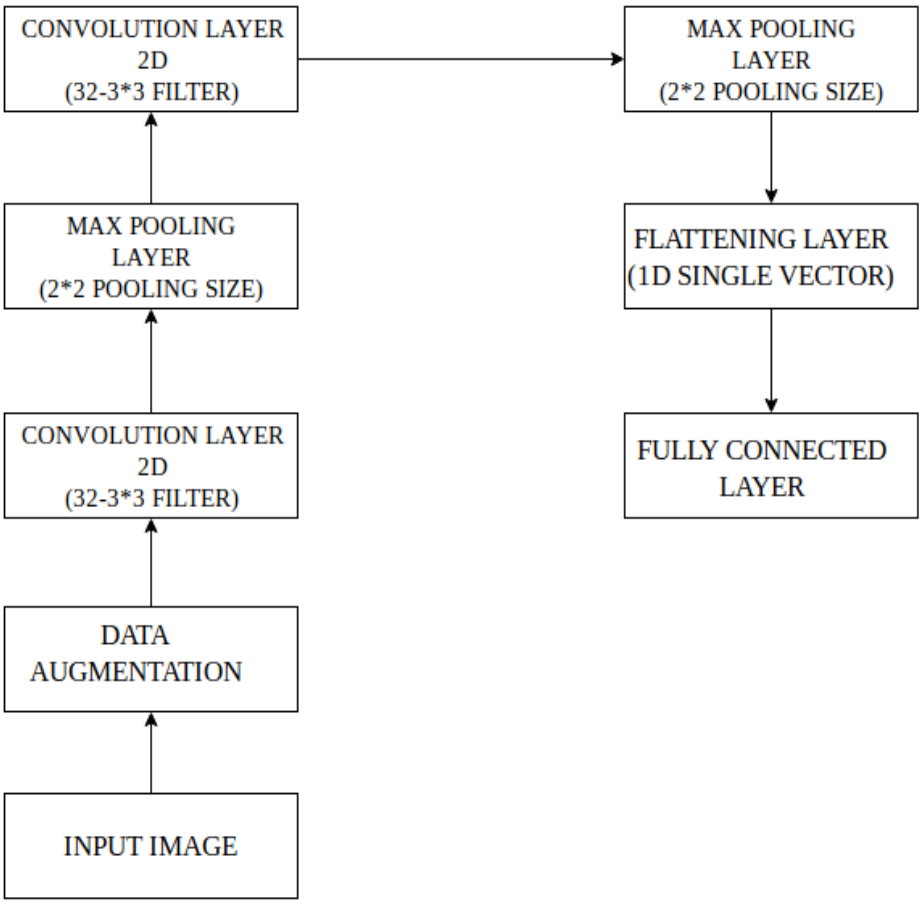

FIGURE 5.11: Network Architecture for glaucoma

designed. The model is trained over 30 epochs. The training model is shown in Figure 5.12.

```
Epoch 1/30
4/3 [==============================] - 50s 12s/step - loss: 2.3300 - acc: 0.6484 - val_loss: 0.5118 - val_acc: 0.8125
Epoch 2/30
4/3 [==============================] - 42s 11s/step - loss: 0.9356 - acc: 0.6562 - val_loss: 0.4280 - val_acc: 0.8438
Epoch 3/30
4/3 [==============================] - 45s 11s/step - loss: 0.5418 - acc: 0.7188 - val_loss: 0.4449 - val_acc: 0.8125
Epoch 4/30
4/3 [==============================] - 44s 11s/step - loss: 0.4446 - acc: 0.8281 - val_loss: 0.3956 - val_acc: 0.8438
Epoch 5/30
4/3 [==============================] - 44s 11s/step - loss: 0.4567 - acc: 0.7578 - val_loss: 0.3519 - val_acc: 0.7812
Epoch 6/30
4/3 [==============================] - 45s 11s/step - loss: 0.4654 - acc: 0.7734 - val_loss: 0.3996 - val_acc: 0.8125
Epoch 7/30
4/3 [==============================] - 43s 11s/step - loss: 0.5190 - acc: 0.7656 - val_loss: 0.4094 - val_acc: 0.6875
Epoch 8/30
4/3 [==============================] - 44s 11s/step - loss: 0.3786 - acc: 0.8203 - val_loss: 0.3892 - val_acc: 0.7812
Epoch 9/30
4/3 [==============================] - 47s 12s/step - loss: 0.3839 - acc: 0.8828 - val_loss: 0.1594 - val_acc: 1.0000
Epoch 10/30
4/3 [==============================] - 44s 11s/step - loss: 0.4481 - acc: 0.8047 - val_loss: 0.4691 - val_acc: 0.7500
Epoch 11/30
4/3 [==============================] - 42s 11s/step - loss: 0.4311 - acc: 0.7891 - val_loss: 0.3674 - val_acc: 0.8750
Epoch 12/30
4/3 [==============================] - 45s 11s/step - loss: 0.2559 - acc: 0.8750 - val_loss: 0.3250 - val_acc: 0.9062
Epoch 13/30
4/3 [==============================] - 45s 11s/step - loss: 0.5409 - acc: 0.8281 - val_loss: 0.3983 - val_acc: 0.8750
Epoch 14/30
4/3 [==============================] - 44s 11s/step - loss: 0.3766 - acc: 0.7891 - val_loss: 0.2936 - val_acc: 0.9062
Epoch 15/30
4/3 [==============================] - 42s 11s/step - loss: 0.3492 - acc: 0.8359 - val_loss: 0.5283 - val_acc: 0.7500
Epoch 16/30
4/3 [==============================] - 47s 12s/step - loss: 0.3512 - acc: 0.8438 - val_loss: 0.3117 - val_acc: 0.8438
Epoch 17/30
4/3 [==============================] - 43s 11s/step - loss: 0.4266 - acc: 0.7969 - val_loss: 0.2703 - val_acc: 0.8750
Epoch 18/30
4/3 [==============================] - 46s 11s/step - loss: 0.3687 - acc: 0.8438 - val_loss: 0.3741 - val_acc: 0.8125
Epoch 19/30
4/3 [==============================] - 42s 11s/step - loss: 0.1969 - acc: 0.9141 - val_loss: 0.2558 - val_acc: 0.8750
Epoch 20/30
4/3 [==============================] - 45s 11s/step - loss: 0.3381 - acc: 0.8594 - val_loss: 0.3588 - val_acc: 0.8125
Epoch 21/30
4/3 [==============================] - 35s 9s/step - loss: 0.4771 - acc: 0.8457 - val_loss: 0.2947 - val_acc: 0.8750
Epoch 22/30
4/3 [==============================] - 2s 511ms/step - loss: 0.2687 - acc: 0.9062 - val_loss: 0.5696 - val_acc: 0.7500
Epoch 23/30
4/3 [==============================] - 2s 594ms/step - loss: 0.4321 - acc: 0.8359 - val_loss: 0.3046 - val_acc: 0.9375
Epoch 24/30
4/3 [==============================] - 2s 612ms/step - loss: 0.2365 - acc: 0.9219 - val_loss: 0.4210 - val_acc: 0.8750
Epoch 25/30
4/3 [==============================] - 2s 611ms/step - loss: 0.2875 - acc: 0.8672 - val_loss: 0.1772 - val_acc: 0.9688
Epoch 26/30
4/3 [==============================] - 3s 636ms/step - loss: 0.2057 - acc: 0.9297 - val_loss: 0.2435 - val_acc: 0.9091
Epoch 27/30
4/3 [==============================] - 3s 704ms/step - loss: 0.2259 - acc: 0.9297 - val_loss: 0.1328 - val_acc: 0.9688
Epoch 28/30
4/3 [==============================] - 2s 594ms/step - loss: 0.2370 - acc: 0.9062 - val_loss: 0.1383 - val_acc: 0.9375
Epoch 29/30
4/3 [==============================] - 2s 593ms/step - loss: 0.1966 - acc: 0.9141 - val_loss: 0.2878 - val_acc: 0.9062
Epoch 30/30
4/3 [==============================] - 2s 600ms/step - loss: 0.1654 - acc: 0.9453 - val_loss: 0.1993 - val_acc: 0.9062
```

FIGURE 5.12: Training Model for Glaucoma

Among 811 testing images the overall accuracy is predicted as 94% which is shown in Figure 5.13.

```
Total:  811
Loss:  0.1728441755387257 Accuracy:  0.9408138101109741
```

FIGURE 5.13: Overall Accuracy for Glaucoma

The sample test image for both glaucoma and not glaucoma detection is shown in Figure 5.14 and Figure 5.15.

```
import numpy as np
from keras.preprocessing import image
test_image = image.load_img('/content/drive/My Drive/glaucom/test/class1/img_0_1031.jpeg', target_size = (256,256))
test_image = image.img_to_array(test_image)
test_image = np.expand_dims(test_image, axis = 0)
result = classifier.predict(test_image)
training_set.class_indices
if result[0][0] == 1:
 print("Glaucoma")
else:
 print("Not Glaucoma")
```

```
Glaucoma
```

FIGURE 5.14: Sample Image Testing for Glaucoma (class1)

```
import numpy as np
from keras.preprocessing import image
test_image = image.load_img('/content/drive/My Drive/glaucom/test/class0/img_0_1016.jpeg', target_size = (256,256))
test_image = image.img_to_array(test_image)
test_image = np.expand_dims(test_image, axis = 0)
result = classifier.predict(test_image)
training_set.class_indices
if result[0][0] == 1:
 print("Glaucoma")
else:
 print("Not Glaucoma")
```

```
Not Glaucoma
```

FIGURE 5.15: Sample Image Testing for Non Glaucoma (class0)

The graph for the detailed model for epochs vs accuracy is shown in figure 5.16.

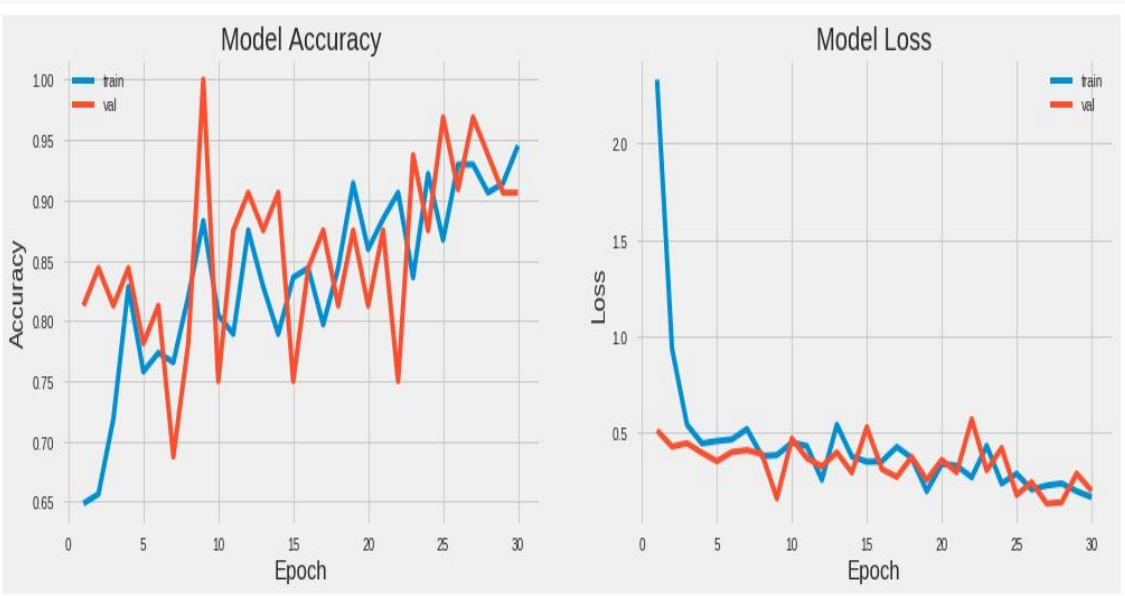

FIGURE 5.16: Graph for the Overall Glaucoma Detection

5.4 Summary

This chapter depicts the implementation of a convolutional neural network for the classification of the Diabetic Retinopathy and detection of Glaucoma. This chapter contains results for both the datasets. The testing accuracy for Diabetic Retinopathy is 86% and it is successfully classified into five different classes. The testing accuracy for glaucoma is 94% and it is successfully predicted into respective classes.

CHAPTER 6

CONCLUSION AND FUTURE WORK

Among other existing supervising algorithms, most of them are requiring more pre-processing or post-processing stages for identifying the different stages of the diabetic retinopathy. Also, other algorithms mandatorily requiring manual feature extraction stages to classify the fundus images. In our proposed solution, Deep convolutional Neural Network (DCNN) is a wholesome approach to all level of diabetic retinopathy stages. No manual feature extraction stages are needed. Our network architecture with dropout techniques yielded significant classification accuracy. Our network architecture is complex and computation-intensive requiring graphics processing unit to process the fundus images when the level of layers stacked more. The testing accuracy of DR can be increased by increasing the number of images in each class and by increasing the convolutional layers. Classification of glaucoma can also be added.

REFERENCES

1. Abhishek, D. and Bandyopadhyay, S. (2016) 'Automated Glaucoma Detection Using Support Vector Machine Classification Method', British Journal of Medicine Medical Research, Vol. 11, No. 12, pp. 1368-1372.

2. Almazroa, A., Burman, R., Raahemifar, K. and Lakshminarayanan, S. (2017) 'Optic Disc and Cup Segmentation Methods for Glaucoma Detection with Modification of U-Net Convolutional Neural Network', Journal of Ophthalmology, Vol. 27, No. 3, pp. 618-624.

3. Baidaa Al-Bander, Waleed Al-Nuaimy and Majid Al-Taee, A. (2017) 'Automated Glaucoma Diagnosis using Deep Learning Approach', 14th International Multi-Conference on Systems, Signals & Devices (SSD), pp. 5386-3175.

4. Burlina, P., Josh, N., Gao, M. and Pekala, M. (2017) 'Automated Grading of Age-Related Macular Degeneration From Color Fundus Images Using Deep Convolutional Neural Networks', The Journal of the American Medical Association Ophthalmology, Vol. 135, No. 11, pp. 1170-1176.

5. Chandrakumar, T. and Kathirvel, R. (2016) 'Classifying Diabetic Retinopathy using Deep Learning Architecture', International Journal of Engineering Research Technology, Vol. 5, No. 6, pp. 2278-0181.

6. Eman Shahin, M., Taha, E., Gao, M. and Al-Nuaimy, W. (2013) 'Automated Detection of Diabetic Retinopathy in Blurred Digital Fundus Images', 52nd Annual Conference of the British Institute for Non-Destructive Testing, pp. 20-25.

7. Giraddi, S. and Pujari, J. (2015) 'Identifying abnormalities in the Retinal Images using SVM Classifiers', International Journal of Computer Applications, Vol. 111, pp. 0975-8887.

8. Gulshan, V., Peng, L., Coram, M. and Narayanaswamy, A. (2016) 'Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs', The Journal of the American Medical Association, Vol. 316, No. 22, pp. 2402-2410.

9. Harry, P., Coenenb, F.,Hardinga, S. and Zhenga, Y. (2017) 'Convolutional Neural Networks for Diabetic Retinopathy', The Journal of Machine Learning Research, Vol. 90, pp. 200-205.

10. Mahendran, G. and Dhanasekaran, R. (2016) 'Diagnosis of Diabetic Retinopathy Using Morphological Process and SVM Classifier', International Conference on Communications and Signal Processing (ICCSP), pp. 873-877.

11. Maheshwari, S., Pachori, R. and Acharya, R. (2017) 'Automated Diagnosis of Glaucoma Using Empirical Wavelet Transform and Correntropy Features Extracted From Fundus Images', IEEE Journal of Biomedical and Health Informatics, Vol. 21, No. 3, pp. 803-813.

12. Nitish, S.,Hinton, G. and Krizhevsky, A. (2014) 'Dropout: A simple way to prevent neural networks from overfitting', The Journal of Machine Learning Research, Vol. 15, No. 1, pp. 1929-1958.

13. Rathinam, S. and Selvarajan, S. (2013) 'Comparison of Image Preprocessing Techniques on Fundus Images for Early Diagnosis of Glaucoma', International Journal of Scientific Engineering Research, Vol. 4, No. 12, pp. 1368-1372.

14. Renuka, M. and HariniPriyaDharsini, B. (2016) 'Analysis of Retinal Blood Vessel using Image Processing techniques', International Conference on Intelligent Computing Applications (ICICA), Vol. 59, pp. 244-248.

15. Roychowdhury, S., Koozekanani, D. and Parhi, K. (2014) ‘Diabetic Retinopathy Analysis Using Machine Learning’, IEEE Journal of Biomedical and Health Informatics, Vol. 18, No. 5, pp. 1717-1728.

16. Sahithya, P., Chitra, P. and Mahalakshmi, K. (2015) ‘A survey based on glaucoma detection’,Imperial Journal of Interdisciplinary Research (IJIR), Vol. 6, pp. 1605-1611.

17. Shin, H., Roth, H., Gao, M. and Summers, S. (2017) ‘Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning’, IEEE Transactions on Medical Imaging, Vol. 35, No. 5, pp. 1285-1298.

18. ‘Diabetic Retinopathy dataset Kaggle’, https://www.kaggle.com/c/classroom-diabetic-retinopathy-detection-competition/data, December 28, 2018.

19. ‘Glaucoma Dataset RIM-ONE’, https://medimrg.webs.ull.es/research/retinal-imaging/rim-one, January 27, 2019.