

GLAUCOMA DETECTION USING CONVOLUTIONAL NEURAL NETWORK

Submitted in partial fulfilment of the required for the award of

Masters of Computer Science

By

BALA VIVETHA V M

REGISTER NO (42734004)



DEPARTMENT OF COMPUTER SCIENCE

SCHOOL OF SCIENCE AND HUMANITIES

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A++” by NAAC

JEPPIAR NAGAR, RAJIV GANDHI SALAI, CHENNAI 600 119

APRIL 2024



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

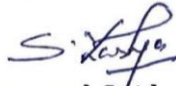
Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **BALA VIVETHA VM (42734004)** who carried out the project entitled "**GLAUCOMA DETECTION USING CONVOLUTIONAL NEURAL NETWORK**" under our supervision from JANUARY 2024 TO APRIL 2024.


Internal Guide

Ms. YUVA SNEHA M.Sc., Assistant Professor

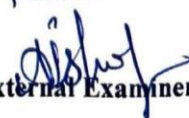

DEAN AND HEAD OF THE DEPARTMENT

Dr. REKHA CHAKRAVARTHI M.E., Ph.D.,

Submitted for Viva Voice examination held on

20/04/2024


Internal Examiner


External Examiner

DECLARATION

I am, BALAVIVETHA V M (42734004) hereby declare that the project report entitled **GLAUCOMA DETECTION USING CONVOLUTIONAL NEURAL NETWORK** done by me under the guidance of **MS. YUVA SNEHA S M.Sc., Assistant Professor**, Department of Computer Science at **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY**, Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai-600 119 is submitted in partial fulfilment of the requirements for the award of Master of Science degree in Computer Science.

DATE: 20.04.2024

PLACE: CHENNAI

BALAVIVETHA V M

SIGNATURE OF THE CANDIDATE



ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA INSTITUTE SCIENCE AND TECHNOLOGY OF** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. REKHA CHAKRAVARTHI M.E., Ph.D., Dean of Arts and Science** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my project guide **MS. YUVA SNEHA S M.Sc., Assistant Professor**, Department of Computer Science for their valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff member of the Department of Computer Science who were helpful in many ways for the completion of the project.

BALAVIVETHA V M

ABSTRACT

Glaucoma is a disease that relates to the vision of the human eye. This disease is considered as the irreversible disease that results in the vision deterioration. Much deep learning (DL) models have been developed for the proper detection of glaucoma so far. So this paper presents architecture for the proper glaucoma detection based on the deep learning by making use of the convolutional neural network (CNN). The differentiation between the patterns formed for glaucoma and non-glaucoma can find out with the use of the CNN. The CNN provides a hierarchical structure of the images for differentiation. Proposed work can be evaluated with a total of six layers. Here the dropout mechanism is also used for achieving the adequate performance in the glaucoma detection. The datasets used for the experiments are the SCES and ORIGA. The analysis is performed for both the dataset and the obtained values are .822 and .882 for the ORIGA and SCES dataset respectively.

Keywords: Glaucoma Detection, Deep Learning, Convolutional Neural Network, Glaucoma Prediction.

TABLE OF CONTENTS

S.NO	CHAPTERS	PAGE NO
1	PROJECT DESCRIPTION	
1.1	Introduction	1
1.2	Glaucoma Detection	1
1.3	Machine Learning	3
1.2.1	Key Consideration	5
1.4	Supervised Machine Learning	6
1.5	Convolutional Neural Network	9
2	LITERATURE SURVEY	10-13
3	PROBLEM SURVEY	
3.1	Existing work	14
3.2	Proposed work	14
4	DROPOUT AND AUGMENTATION	
4.1	Criteria for the Evolution	16
4.2	Setup for the Experiment	16
5	LOGICAL DEVELOPMENT	
5.1	DFDS	17
5.2	Architectural Design	18
5.3	Layers of CNN Architecture	19
6	MODEL DESIGN	21-36
7	PROGRAM DESIGN	
7.1	Modules	37-38
8	TESTING	39-41
9	CONCLUSION	42
10	APPENDIX	
10.1	Source Code	43-46
10.2	Sample Input and Output	47-48
11	REFERENCE	49-51

LIST OF FIGURES

LIST OF FIGURES	NAME OF THE FIGURE	PAGE NO
1.1	Image for the Infected Glaucoma Eye	3
1.2	Supervised learning in ML	7
5.1	Data Flow Design	17
5.2	Architectural Design	18
5.3	Layers of CNN Architecture	19
7.1	Data processing	38
8	Testing(output)	40-41
10	Sample Input & Output	47-48

1. PROJECT DESCRIPTION

1.1 INTRODUCTION

Glaucoma is one of the critical diseases that may lead to the blind vision of the person. Glaucoma is one of the cases in which the optic nerve of the eye gets affected, and this is the main reason for the vision loss. The first operation of glaucoma of human was done in 1856 by Graefe [1]. The whole population that is dealing with the problem of glaucoma may lose their vision without proper treatment and the care for that. The specialist in eye care can find the patients who suffer from this disease [1]. Glaucoma internally contains various conditions that have similar characteristics. There are many works done in this field for the early detection of this disease. The system used various Deep learning Algorithms for proper detection. As stated, that the early detection can prevent the blindness in human being and the vision can be saved. So, the appropriate detecting model is required for the detection of this disease. There are many attempts taken for developing such a system. The approach is also presented to detect the glaucoma pattern in the patients. The presented method will make the use of CNN technique for the classification of the patterns found in patients. The differentiation of the trends in the founded data for the glaucoma detection will be carried out by making use of the CNN model. The overall architecture is having six layers for proper detection of the disease. In the presented mechanism, a dropout mechanism is also applied to improve the performance of the given approach. The major objective is to find the most similar patterns in between the normal human eye and the infected glaucoma eye.

1.2 GLAUCOMA DETECTION

Glaucoma is the disease that affects the human eye and results in a permanent blindness in the human eye. This situation is seen as the complex to deal with, so the proper detection is a must. This problem is detected at an early stage then it may be improved else it may lead to loss of vision.

As per the previous evaluation, a single checkup cannot find the symptoms of glaucoma. The regular eye checks up may give the symptoms of glaucoma, and further treatment and check-up may be suggested. The eye specialist checks five times at least for the confirmation of this disease in the human eye. Following are some health diagnoses that is analyzed for the approval of glaucoma.

Tonometry:

Tonometry is a diagnostic test that measures the pressure inside your eye, which is called intraocular pressure (IOP). This measurement can help your doctor determine whether or not you may be at risk of glaucoma. Glaucoma is a serious eye disease that can eventually lead to vision loss if untreated.

Optical Coherence Tomography:

This scan is essential for the diagnosis of glaucoma. It is used to find an important sign of early glaucoma damage that is retinal nerve fiber layers around the optic nerve.

Ophthalmoscopy:

The optic nerve is examined in this test. As glaucoma is a severe disease related to the optic nerve, so this is a very important test. Eye drops are used to enlarge the size of the pupil of the patient's eye to look optic nerve more clearly to find signs of disease-related nerve cell loss in the eye.

Perimetry:

Glaucoma is a disease which causes peripheral vision loss at the initial stage. Therefore, this test is done to detect vision loss. This test is also called a visual field test. It includes testing each eye distinctly with an automatic device that flashes lights in the periphery of the eye of the person.

Gonioscopy:

It is the test related to intraocular fluid outflow drainage angle. The fluid is continuously being prepared in the eye & then it flows out at fixed angles. This test is done to find that whether the high eye pressure is caused by a blocked angle that is known as angle-closure glaucoma or if angle is open but not working properly then it is known to be as open-angle glaucoma.

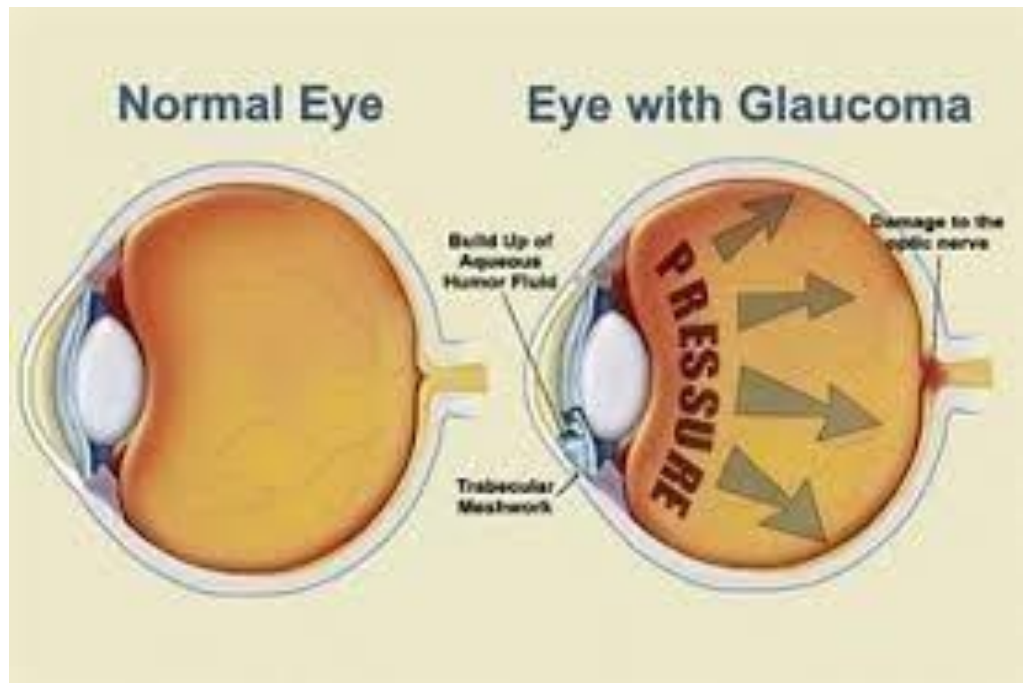


Figure 1.1 Image for the Infected Glaucoma Eye.[1]

1.3 MACHINE LEARNING

Machine learning (ML) is the study of computer algorithms that improve automatically through experience. Machine learning algorithms build a model based on sample data, known as "training data", in order to make predictions or decisions. ML gives the computer that makes it more similar to humans: The ability to learn. The basic process of machine learning is to give training data to a learning algorithm. The learning algorithm then generates a new set of rules, based on inferences from the data. This is in essence generating a new algorithm, formally referred to as the machine learning model. By using different training data, the same learning algorithm could be used to generate different models. For example, the same type of learning algorithm could be used to teach the computer how to translate languages or predict the stock market. Inferring new instructions from data is the core strength of machine learning. It also highlights the critical role of data: the more data available to train the algorithm, the more it learns. In fact, many recent advances in AI have not been due to radical innovations in learning algorithms, but rather by the enormous amount of data enabled by the Internet. Although a machine learning model may apply a mix of different techniques, the methods for learning can typically be categorized as three general types:

Supervised Learning:

The learning algorithm is given labeled data and the desired output. For example, pictures of dogs labeled “dog” will help the algorithm identify the rules to classify pictures of dogs.

Regression, Decision Tree, Random Forest.

Unsupervised Learning:

The data given to the learning algorithm is unlabeled, and the algorithm is asked to identify patterns in the input data. For example, the recommendation system of an e-commerce website where the learning algorithm discovers similar items often bought together.

Apriori algorithm, K means reinforcement Learning.

The algorithm interacts with a dynamic environment that provides feedback in terms of rewards and punishments. For example, self-driving cars being rewarded to stay on the road.

Markov Decision Process.

Machine learning is tied to developments in three important areas:

Data availability:

Just over 3 billion people are online with an estimated 17 billion connected devices or sensors. That generates a large amount of data which, combined with decreasing costs of data storage, is easily available for use. Machine learning can use this as training data for learning algorithms, developing new rules to perform increasingly complex tasks.

Computing power:

Powerful computers and the ability to connect remote processing power through the Internet make it possible for machine-learning techniques that process enormous amounts of data.

Algorithmic innovation:

New machine learning techniques, specifically in layered neural networks – also known as “deep learning” – have inspired new services, but is also spurring investments and research in other parts of the field.

1.2.1 KEY CONSIDERATION:

As machine learning algorithms are used in more and more products and services, there are some serious factors must be considered when addressing AI, particularly in the context of people's trust in the Internet:

Socio-economic impacts.

The new functions and services of AI are expected to have significant socio-economic impacts. The ability of machines to exhibit advanced cognitive skills to process natural language, to learn, to plan and to perceive, makes it possible for new tasks to be performed by intelligent systems, sometimes with more success than humans. New applications of AI could open up exciting opportunities for more effective medical care, safer industries and services, and boost productivity on a massive scale.

Transparency, bias and accountability.

AI-made decisions can have serious impacts in people's lives. AI may discriminate against some individuals or make errors due to biased training data. How a decision is made by AI is often hard to understand, making problems of bias harder to solve and ensuring accountability much more difficult.

New uses for data.

Machine learning algorithms have proved efficient in analyzing and identifying patterns in large amounts of data, commonly referred to as "Big Data". Big Data is used to train learning algorithms to increase their performance. This generates an increasing demand for data, encouraging data collection and raising risks of oversharing of information at the expense of user privacy.

Security and safety.

Advancements in AI and its use will also create new security and safety challenges. These include unpredictable and harmful behavior of the AI agent, but also adversarial learning by malicious actors.

Ethics.

AI may make choices that could be deemed unethical, yet also be a logical outcome of the algorithm, emphasizing the importance to build in ethical considerations into AI systems and algorithms.

New ecosystems.

Like the impact of mobile Internet, AI makes new applications, services, and new means of interacting with the network possible. For example, through speech and smart agents, which may create new challenges to how open or accessible the Internet becomes.

1.4 SUPERVISED MACHINE LEARNING

- Majority of practical machine learning uses supervised learning.
- Supervised learning is the types of machine learning in which machines are trained using well "labelled" training data, and on basis of that data, machines predict the output. The labelled data means some input data is already tagged with the correct output.
- In supervised learning, the training data provided to the machines work as the supervisor that teaches the machines to predict the output correctly. It applies the same concept as a student learns in the supervision of the teacher.
- Supervised learning is a process of providing input data as well as correct output data to the machine learning model. The aim of a supervised learning algorithm is to **find a mapping function to map the input variable(x) with the output variable(y).**
- Supervised learning is where you have input variables (x) and an output variable (Y) and you use an algorithm to learn the mapping function from the input to the output.
 $Y = f(X)$.
- Goal is to approximate the mapping function with new input data (x) that can predict the output variables (Y) for that data. In the real-world, supervised learning can be used for **Risk Assessment, Image classification, Fraud Detection, spam filtering**, etc.

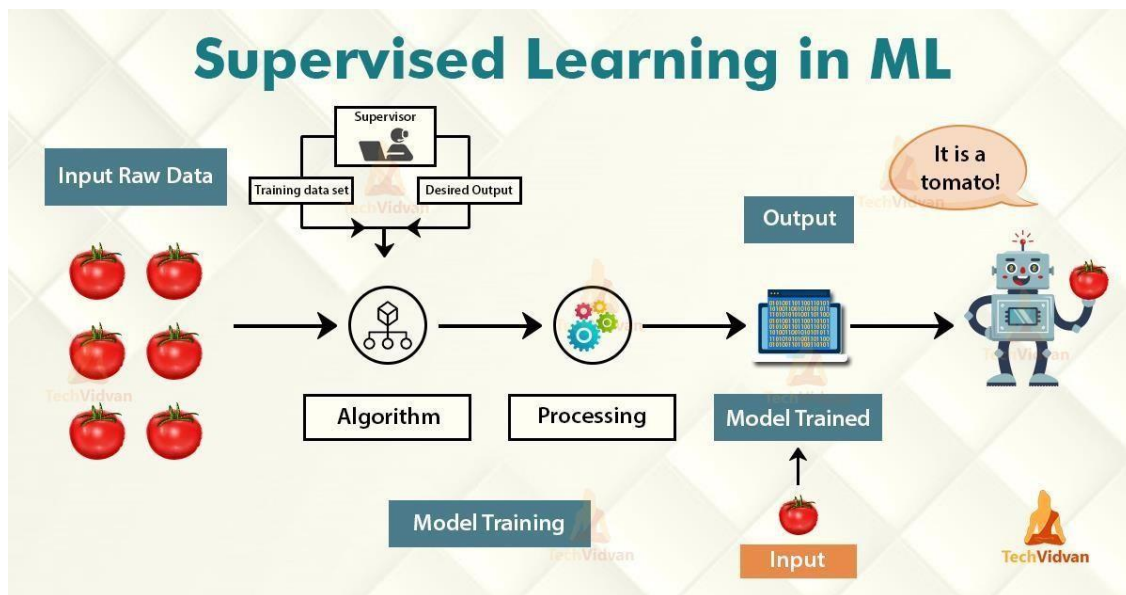


Figure 1.2 Supervised learning in ML

Steps involved in supervised machine learning algorithms:

- First Determine the type of training dataset □ Collect/Gather the labelled training data.
- Split the training dataset into training **dataset**, **test dataset**, and **validation dataset**.
- Determine the input features of the training dataset, which should have enough knowledge so that the model can accurately predict the output.
- Determine the suitable algorithm for the model, such as support vector machine, decision tree, etc.
- Execute the algorithm on the training dataset. Sometimes we need validation sets as the control parameters, which are the subset of training datasets.
- Evaluate the accuracy of the model by providing the test set. If the model predicts the correct output, which means our model is accurate.

Types of supervised Machine learning Algorithms:

Supervised learning can be further divided into two types of problems:

1. Regression:

Regression algorithms are used if there is a relationship between the input variable and the output variable. It is used for the prediction of continuous variables, such as Weather forecasting,

Market Trends, etc. Below are some popular Regression algorithms which come under supervised learning:

- Linear Regression
- Regression Trees
- Non-Linear Regression
- Bayesian Linear Regression □ Polynomial Regression.

2. Classification

Classification algorithms are used when the output variable is categorical, which means there are two classes such as Yes-No, Male-Female, True-false, etc.

- Random Forest
- Decision Trees
- Logistic Regression
- Support vector Machines

Advantages:

- Supervised learning allows collecting data and produces data output from previous experiences.
- Helps to optimize performance criteria with the help of experience.
- Supervised machine learning helps to solve various types of real-world computation problems.
- With the help of supervised learning, the model can predict the output on the basis of prior experiences.
- In supervised learning, we can have an exact idea about the classes of objects.
- Supervised learning model helps us to solve various real-world problems such as **fraud detection, spam filtering**, etc.

1.5 CONVOLUTIONAL NEURAL NETWORK

Convolutional neural network (ConvNets or CNNs) is one of the main categories to perform image recognition and image classification.

CNN image classifications:

- Takes an input image.
- Process it.
- Classifies it under certain categories (Glaucoma or Non-Glaucoma).

The main advantage of CNN compared to its predecessors is that it automatically detects the important features without any human supervision. CNN architecture used in proposed system is DenseNet121.

2. LITERATURE SURVEY

Rashmi Panda et al. [1] put forward an automated model for Retinal nerve fiber layer defect detection. As it is an early proof of glaucoma condition in fundus images. Early detection & prevention are ways to stop loss of vision. The new method performs detection in fundus images using patch characteristics driven RNN. Fundus images dataset is used to evaluate performance. This system obtains high RNFLD detection and accurate boundary localization.

Kavi ta Choudhary et al. [2] presented a paper with the aim of detection of glaucoma at early stages using cross-validation algorithm. Authors analyzed symptoms prevailing in persons & computed & generalized those symptoms to reach conclusive evidence. It was found that measures such as blood pressure, Age, Sugar level, & myopia were combined for various datasets are related to changes of a person suffering from glaucoma. Authors in their study have done an analysis of glaucoma disease by Classification method such as cross-validation algorithm & split validation algorithm. The outcome reveals that patients who have high blood pressure, high sugar level, myopia & with the family history of this disease can suffer from glaucoma. It is also observed that the patients with age more than 50 have higher chances of glaucoma.

Seong Jae Kim et al. [3] in this paper studied and attempted to design machine learning models that have robust power of predicting & interpretability for glaucoma diagnosis based on RNFL thickness & visual field. Different features were collected after examination of RNFL thickness & visual field. Authors used 4 machine learning algorithms like C5.0, random forest, SVM & K-nearest neighbor to design a glaucoma prediction model. Learning models are constructed using training dataset & their performance was evaluated by using validation data set. Finally, the authors observed that the random forest model gives the best performance & remaining other models show similar accuracy.

Shwetha C. Shetty et al. [4] discussed and analyzed that Glaucoma is an ocular disorder and its identification includes measuring the shapes and also the optic cup sizes. Pre-processing of data is then clustered using K-means clustering, which is used for segmenting the optic curves. It is again executed to find its various dimension. Since the fractional dimension is used to determine the various dimension of non-regular identities, the authors presented a new method for detection of glaucoma using method of the perimeter for the fractional analysis. The outcome reveals the new approach is accurate in detecting glaucoma.

Liuli et al. [5] in this paper presented attention-based convolutional neural network for detecting glaucoma, known as AGCNN. Approaches which were proposed in the past for automatic detection system based on fundus images are insufficient to remove high redundancy, which may lead to reduced reliability & accuracy of detection. To overcome this shortcoming, the newly proposed method establishes a large-scale data set, which includes fundus images labelled as (+) ve or (–) ve. The attention maps of some images are taken from ophthalmologists through a simulated experiment. Then a new AG- CNN structure is constructed which includes a subnet, a pathological area localization subnet and a glaucoma classification subnet. Experiment on LAG database & other available datasets reveals that the proposed method gives a detection performance superior to previous models.

JinMo Ahn et al. [6] here, they have presented a method for the detection of the glaucoma disease which utilizes fundus photography and uses deep learning. The author discussed that advanced & early glaucoma both could be correctly identified using machine learning along with fundus images. Dataset of 1,542 images was used and divided into training, validation & test datasets. The newly put forward model that is trained using CNN is more effective and accurate in the detection of early glaucoma.

Annan Li et al. [7] suggest that automatic detection of disease is important for retinal image analysis. When studied and compared with segmentation-based approaches, it is found that image classification-based approaches perform better. But challenges are always there due to improper sample, effective features and also shape variations of the optic disc. To overcome these, a new classification-based model for detection of glaucoma is put forward

by authors in this paper, in which deep convolutional networks are used to represent visual appearance, holistic & local characteristics are combined to reduce or remove misalignment.

Ali Serener et al. [8] discussed Open-angle glaucoma as it is one of the basic kinds of disease & slowly, a person tends to lose his sight. Diagnosis of this disease manually by experts is possible, but it either takes a huge time or cost. Author sin this paper presented a method for the detection of both early &advanced glaucoma automatically. ‘ResNet-50’ & ‘Google Net’ deep CNN algorithms are trained &tuned using transfer learning. It is found that ‘GoogLeNet’ model is better than ‘ResNet-50’ for detecting both early &advanced glaucoma in the eye of the patient.

RaminDaneshvaret al. [9] analyzed that baseline OCT measures predict VF progression in patients with suspicious or established glaucoma & also authors compared performance with semi- quantitative optic disc measures. It is observed that baseline PRNFL & macular OCT parameters can be used for checking the risk of glaucoma progression in future. People abnormal OCT findings require better care to prevent progression of functional damage.

Guangzhou An et al. [10] presented a model for the detection of glaucoma within the patients by making the use of the open-angle for glaucoma that is based on the 3-D data color images. Various fundus pictures as input provide the CNN architecture. After getting output form every CNN model, the outputs have been combined. And the random forest method is used for the classification of the fundus pictures. This classification is done with healthy and glaucoma infected eyes. At the result obtained for the AUC is of.96.

Juan Carrillo et al. [11] The authors have provided the glaucoma detection method as the glaucoma is the irreversible cure of eyes. They have provided a tool for computing the glaucoma symptoms in eyes. They have used this tool for the detection, and the detection is observed by the sizes of the cup and the disc. Also, they have used the fundus images for the evaluation.

Tehmina Khalil et al. [12] have presented an overview of the glaucoma detection, and they have stated that most of the detection schemes for glaucoma use the fundus images. They stated that the detection of glaucoma could be done efficiently by making the use of the Optical Coherence Tomography. By using the OCT, the detection can be done at an early stage is concluded by them.

Namita Sengar et al. [13] have stated that the image processing for glaucoma detection can be done by using the fundus images. They have presented a deciding parameter for the detection of glaucoma. The work proposed by them performed well, and their presented mechanism has achieved an accuracy of 93.57%.

Abdul basher et al. [14] presented a survey on the deep learning neural networks. They stated that the neural networks do the training to the system. They stated about the CNN that the feature extraction in CNN is done automatically that it reduces the extra work of feature extraction.

T. Vijayakumar et al. [15] have presented a capsule neural network. This study compared the drawbacks of CNN with the capsule neural network. The study is done on various applications.

3. PROBLEM SURVEY

3.1 EXISTING WORK

There are many existing models for identifying glaucoma. They are:

- Automated diagnosis of glaucoma using empirical wavelet transform and correntropy features extracted from fundus images.□
- Automated Diagnosis of Glaucoma Using texture and higher order spectra features.□
- Existing methods required more improvements for glaucoma detection.□

DISADVANTAGE OF EXISTING WORK

Physician requires at least 5 check-up reports to confirm the glaucoma affection

- Difficult to design a system to detect this disease accurately in single attempt.□
- Lack of accuracy.□

3.2 PROPOSED WORK

In the proposed model we have used certain methods to make it more efficient than the existing:

- Convolutional Neural Network□
- Image Data Generator□
- Denset121□
- Neural Network Classifier□

ADVANTAGES OF PROPOSED WORK

- Feature extraction is obtained by optic cup and optic disc segmentation.□
- Dataset is augmented to improve image classification.□
- Enables model to learn in unsupervised learning□

4. PREREQUISITES

DROPOUT AND DATA AUGMENTATION

Dropout:

It is implemented in two stages of the fully connected layers as presented in our approach. The Dropout will set 0 for every value of neuron, which is having a value of .5. When the dropping of the neurons is done in the CNN, then they will not take part in the passing forward and will not be a part of the backpropagation. While experimenting with the multiplication for all the outputs of the neurons by .5 is done.

Data Augmentation:

The overfitting suffering will be seen if the data augmentation is not done in the model. DA will generate translations for the images and the reflections that are horizontal. For the training period 224×224 patch is done for random values with the inclusion of the 256×256 models, and train the network for these patches that are extracted. In testing five 224×224 patches are obtained from CNN, which include four corners and one center. The horizontal reflections are also obtained for these five patches. The average of this prediction that is given by the network softmax layer for this 10 patches is taken.

DATASET AND INPUT

- Our dataset consists of 2 lakhs images.□
- Among which, 1 lakh images are glaucoma images and 1 lakh images are healthy eye images.□
- 85 percentage of the images is used for training the machine learning model.□
- 15 percentage of the images is used for testing the trained model.□

EXPERIMENTS PERFORMED

The proposed model of two datasets that are ORIGA and SCES, which is having the images of glaucoma fundus are experimented here.

4.1 CRITERIA FOR THE EVALUATION

The area under the curve (AUC) is utilized of the receiver operating characteristics (ROC) curve for the evaluation of glaucoma detecting performance of the model. The curve between the sensitive TPR and the specificity TNR is plotted as the ROC and is defined as:

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \text{ TNR} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

4.2 SETUP FOR THE EXPERIMENT

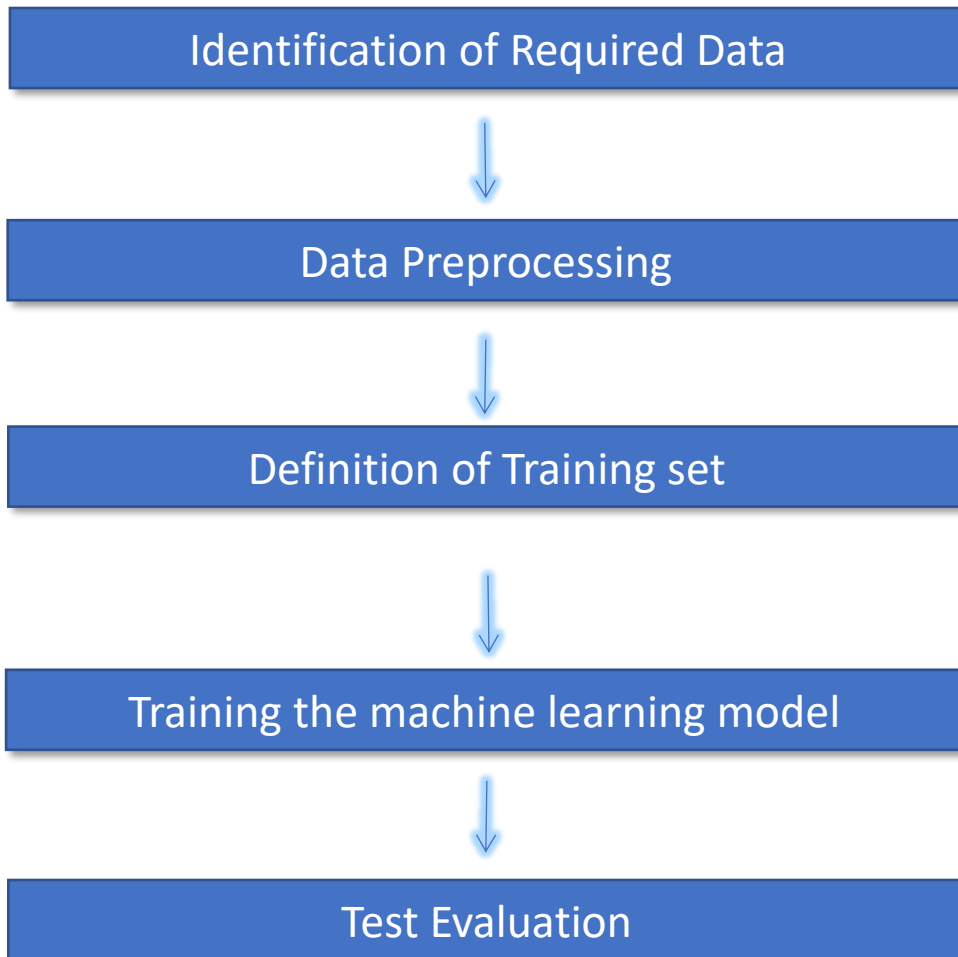
The ORIGA dataset consists of clinical glaucoma diagnoses and having 168 glaucoma and 482 images of the normal fundus. The dataset of the SCES has 46 images for glaucoma and 1676 images for the fundus. The experiment carried out has the following configurations:

- **In Hardware:** 4 GB RAM, 140 Gb SSD Hard disk, Intel core i3 and intel motherboard.
- **In Software:** 64-bit windows 10 and Python3.8.
- **Python packages** like NumPy, pandas, tensor flow and KerasLibrary etc.
- **Data Set:** ORIGA dataset consists of clinical glaucoma diagnoses and having 168 glaucoma and 482 images of the normal fundus.

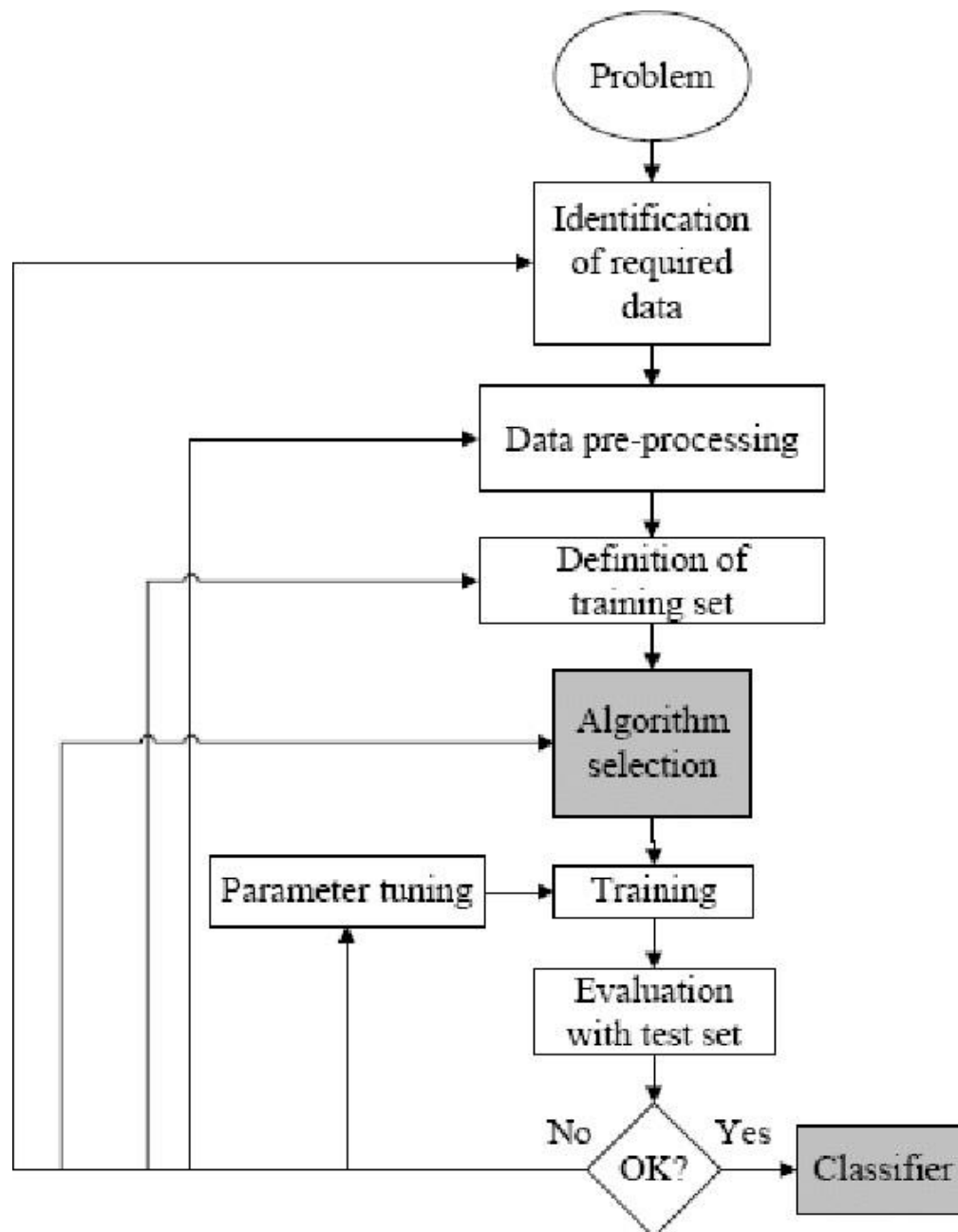
5.LOGICAL DEVELOPMENT

5.1 DFDs.

Data Flow Design



5.2 ARCHITECTURAL DESIGN



5.3 LAYERS OF CNN ARCHITECTURE:

Input layer:

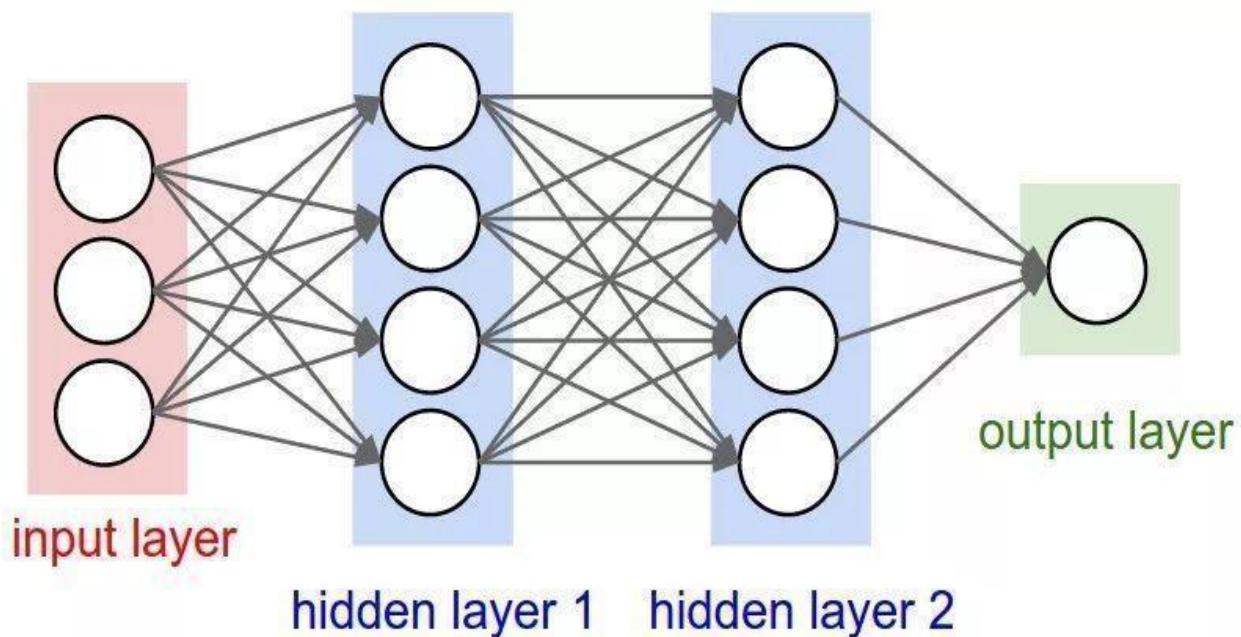
A number of neurons (Neuron) accept a large number of non-linear input messages. The input message is called the input vector.

Output layer:

Here, the message is transmitted, analyzed, and weighed in the neuron link to form an output. The output message is called the output vector.

Hidden layer:

It is referred to as the "hidden layer", is the layer of many neurons and links between the input layer and the output layer. If there are multiple hidden layers, it means multiple activation functions.



CNN Architecture layers

5.4 ADVANTAGES OF CNN:

Below are some applications of Convolutional Neural Networks used today:

1. Object detection:

With CNN, we now have sophisticated models like [R-CNN](#), [Fast R-CNN](#), and [Faster R-CNN](#) that are the predominant pipeline for many object detection models deployed in autonomous vehicles, facial detection, and more.

2. Semantic segmentation:

In 2015, a group of researchers from Hong Kong developed a CNN-based [Deep Parsing Network](#) to incorporate rich information into an image segmentation model. Researchers from UC Berkeley also built [fully convolutional networks](#) that improved upon state-of-the-art semantic segmentation.

3. Image captioning:

CNNs are used with recurrent neural networks to write captions for images and videos. This can be used for many applications such as activity recognition or describing videos and images for the visually impaired. It has been heavily deployed by YouTube to make sense to the huge number of videos uploaded to the platform on a regular basis.

6.MODEL DESIGN

Found 520 images belonging to 2 classes.

Found 130 images belonging to 2 classes.

1024

512

256

{'Glaucoma_Negative': 0, 'Glaucoma_Positive': 1} Model:

"functional_3"

Layer (type)	Output Shape	Param #	Connected to
=====			
=			
===== input_2 (InputLayer)			
	[(None, 300, 300, 3) 0		
conv1_pad (ZeroPadding2D)	(None, 306, 306, 3) 0		input_2[0][0]
conv1_conv (Conv2D)	(None, 150, 150, 64) 9472		conv1_pad[0][0]
conv1_bn (BatchNormalization)	(None, 150, 150, 64) 256		conv1_conv[0][0]
conv1_relu (Activation)	(None, 150, 150, 64) 0		conv1_bn[0][0]
pool1_pad (ZeroPadding2D)	(None, 152, 152, 64) 0		conv1_relu[0][0]
pool1_pool (MaxPooling2D)	(None, 75, 75, 64) 0		pool1_pad[0][0]

conv2_block1_1_conv (Conv2D)	(None, 75, 75, 64) 4160	pool1_pool[0][0]
conv2_block1_1_bn (BatchNormali	(None, 75, 75, 64) 256	conv2_block1_1_conv[0][0]
conv2_block1_1_relu (Activation	(None, 75, 75, 64) 0	conv2_block1_1_bn[0][0]
conv2_block1_2_conv (Conv2D)	(None, 75, 75, 64) 36928	conv2_block1_1_relu[0][0]
conv2_block1_2_bn (BatchNormali	(None, 75, 75, 64) 256	conv2_block1_2_conv[0][0]
conv2_block1_2_relu (Activation	(None, 75, 75, 64) 0	conv2_block1_2_bn[0][0]
conv2_block1_0_conv (Conv2D)	(None, 75, 75, 256) 16640	pool1_pool[0][0]
conv2_block1_3_conv (Conv2D)	(None, 75, 75, 256) 16640	conv2_block1_2_relu[0][0]
conv2_block1_0_bn (BatchNormali	(None, 75, 75, 256) 1024	conv2_block1_0_conv[0][0]
conv2_block1_3_bn (BatchNormali	(None, 75, 75, 256) 1024	conv2_block1_3_conv[0][0]
conv2_block1_add (Add)	(None, 75, 75, 256) 0	conv2_block1_0_bn[0][0] conv2_block1_3_bn[0][0]
conv2_block1_out (Activation)	(None, 75, 75, 256) 0	conv2_block1_add[0][0]
conv2_block2_1_conv (Conv2D)	(None, 75, 75, 64) 16448	conv2_block1_out[0][0]

conv2_block2_1_bn (BatchNormali (None, 75, 75, 64) 256 conv2_block2_1_conv[0][0]

conv2_block2_1_relu (Activation (None, 75, 75, 64) 0 conv2_block2_1_bn[0][0]

conv2_block2_2_conv (Conv2D) (None, 75, 75, 64) 36928 conv2_block2_1_relu[0][0]

conv2_block2_2_bn (BatchNormali (None, 75, 75, 64) 256 conv2_block2_2_conv[0][0]

conv2_block2_2_relu (Activation (None, 75, 75, 64) 0 conv2_block2_2_bn[0][0]

conv2_block2_3_conv (Conv2D) (None, 75, 75, 256) 16640 conv2_block2_2_relu[0][0]

conv2_block2_3_bn (BatchNormali (None, 75, 75, 256) 1024 conv2_block2_3_conv[0][0]

conv2_block2_add (Add) (None, 75, 75, 256) 0 conv2_block1_out[0][0]
conv2_block2_3_bn[0][0]

conv2_block2_out (Activation) (None, 75, 75, 256) 0 conv2_block2_add[0][0]

conv2_block3_1_conv (Conv2D) (None, 75, 75, 64) 16448 conv2_block2_out[0][0]

conv2_block3_1_bn (BatchNormali (None, 75, 75, 64) 256 conv2_block3_1_conv[0][0]

conv2_block3_1_relu (Activation (None, 75, 75, 64) 0 conv2_block3_1_bn[0][0]

conv2_block3_2_conv (Conv2D)	(None, 75, 75, 64)	36928	conv2_block3_1_relu[0][0]
conv2_block3_2_bn (BatchNormali	(None, 75, 75, 64)	256	conv2_block3_2_conv[0][0]
conv2_block3_2_relu (Activation	(None, 75, 75, 64)	0	conv2_block3_2_bn[0][0]
conv2_block3_3_conv (Conv2D)	(None, 75, 75, 256)	16640	conv2_block3_2_relu[0][0]
conv2_block3_3_bn (BatchNormali	(None, 75, 75, 256)	1024	conv2_block3_3_conv[0][0]
conv2_block3_add (Add)	(None, 75, 75, 256)	0	conv2_block2_out[0][0] conv2_block3_3_bn[0][0]
conv2_block3_out (Activation)	(None, 75, 75, 256)	0	conv2_block3_add[0][0]
conv3_block1_1_conv (Conv2D)	(None, 38, 38, 128)	32896	conv2_block3_out[0][0]
conv3_block1_1_bn (BatchNormali	(None, 38, 38, 128)	512	conv3_block1_1_conv[0][0]
conv3_block1_1_relu (Activation	(None, 38, 38, 128)	0	conv3_block1_1_bn[0][0]
conv3_block1_2_conv (Conv2D)	(None, 38, 38, 128)	147584	conv3_block1_1_relu[0][0]
conv3_block1_2_bn (BatchNormali	(None, 38, 38, 128)	512	conv3_block1_2_conv[0][0]

conv3_block1_3_bn (BatchNormali (None, 38, 38, 512) 2048 conv3_block1_3_conv[0][0]

conv3_block1_2_relu (Activation (None, 38, 38, 128) 0 conv3_block1_2_bn[0][0]

conv3_block1_0_conv (Conv2D) (None, 38, 38, 512) 131584 conv2_block3_out[0][0]

conv3_block1_3_conv (Conv2D) (None, 38, 38, 512) 66048 conv3_block1_2_relu[0][0]

conv3_block1_0_bn (BatchNormali (None, 38, 38, 512) 2048 conv3_block1_0_conv[0][0]

conv3_block1_add (Add)	(None, 38, 38, 512) 0	conv3_block1_0_bn[0][0] conv3_block1_3_bn[0][0]
conv3_block1_out (Activation)	(None, 38, 38, 512) 0	conv3_block1_add[0][0]
conv3_block2_1_conv (Conv2D)	(None, 38, 38, 128) 65664	conv3_block1_out[0][0]
conv3_block2_1_bn (BatchNormali	(None, 38, 38, 128) 512	conv3_block2_1_conv[0][0]
conv3_block2_1_relu (Activation	(None, 38, 38, 128) 0	conv3_block2_1_bn[0][0]
conv3_block2_2_conv (Conv2D)	(None, 38, 38, 128) 147584	conv3_block2_1_relu[0][0]
conv3_block2_2_bn (BatchNormali	(None, 38, 38, 128) 512	conv3_block2_2_conv[0][0]
conv3_block2_2_relu (Activation	(None, 38, 38, 128) 0	conv3_block2_2_bn[0][0]
conv3_block2_3_conv (Conv2D)	(None, 38, 38, 512) 66048	conv3_block2_2_relu[0][0]
conv3_block2_3_bn (BatchNormali	(None, 38, 38, 512) 2048	conv3_block2_3_conv[0][0]
conv3_block2_add (Add)	(None, 38, 38, 512) 0	conv3_block1_out[0][0] conv3_block2_3_bn[0][0]
conv3_block2_out (Activation)	(None, 38, 38, 512) 0	conv3_block2_add[0][0]
conv3_block3_1_conv (Conv2D)	(None, 38, 38, 128) 65664	conv3_block2_out[0][0]
conv3_block3_1_bn (BatchNormali	(None, 38, 38, 128) 512	conv3_block3_1_conv[0][0]

conv3_block3_1_relu (Activation (None, 38, 38, 128) 0	conv3_block3_1_bn[0][0]
conv3_block3_2_conv (Conv2D) (None, 38, 38, 128) 147584	conv3_block3_1_relu[0][0]
conv3_block3_2_bn (BatchNormali (None, 38, 38, 128) 512	conv3_block3_2_conv[0][0]
conv3_block3_2_relu (Activation (None, 38, 38, 128) 0	conv3_block3_2_bn[0][0] conv3_block3_3_conv
(Conv2D) (None, 38, 38, 512) 66048	conv3_block3_2_relu[0][0]
conv3_block3_3_bn (BatchNormali (None, 38, 38, 512) 2048	conv3_block3_3_conv[0][0]
conv3_block3_add (Add) (None, 38, 38, 512) 0	conv3_block2_out[0][0]
	conv3_block3_3_bn[0][0]
conv3_block3_out (Activation) (None, 38, 38, 512) 0	conv3_block3_add[0][0]
conv3_block4_1_conv (Conv2D) (None, 38, 38, 128) 65664	conv3_block3_out[0][0]
conv3_block4_1_bn (BatchNormali (None, 38, 38, 128) 512	conv3_block4_1_conv[0][0]
conv3_block4_1_relu (Activation (None, 38, 38, 128) 0	conv3_block4_1_bn[0][0]
conv3_block4_2_conv (Conv2D) (None, 38, 38, 128) 147584	conv3_block4_1_relu[0][0]
conv3_block4_2_bn (BatchNormali (None, 38, 38, 128) 512	conv3_block4_2_conv[0][0]
conv3_block4_2_relu (Activation (None, 38, 38, 128) 0	conv3_block4_2_bn[0][0]

conv3_block4_3_conv (Conv2D)	(None, 38, 38, 512) 66048	conv3_block4_2_relu[0][0]
conv3_block4_3_bn (BatchNormali	(None, 38, 38, 512) 2048	conv3_block4_3_conv[0][0]
conv3_block4_add (Add)	(None, 38, 38, 512) 0	conv3_block3_out[0][0] conv3_block4_3_bn[0][0]
conv3_block4_out (Activation)	(None, 38, 38, 512) 0	conv3_block4_add[0][0]
conv4_block1_1_conv (Conv2D)	(None, 19, 19, 256) 131328	conv3_block4_out[0][0]
conv4_block1_1_bn (BatchNormali	(None, 19, 19, 256) 1024	conv4_block1_1_conv[0][0]
conv4_block1_1_relu (Activation	(None, 19, 19, 256) 0	conv4_block1_1_bn[0][0]
conv4_block1_2_conv (Conv2D)	(None, 19, 19, 256) 590080	conv4_block1_1_relu[0][0]
conv4_block1_2_bn (BatchNormali	(None, 19, 19, 256) 1024	conv4_block1_2_conv[0][0]
conv4_block1_2_relu (Activation	(None, 19, 19, 256) 0	conv4_block1_2_bn[0][0]
conv4_block1_0_conv (Conv2D)	(None, 19, 19, 1024) 525312	conv3_block4_out[0][0]
conv4_block1_3_conv (Conv2D)	(None, 19, 19, 1024) 263168	conv4_block1_2_relu[0][0]
conv4_block1_0_bn (BatchNormali	(None, 19, 19, 1024) 4096	conv4_block1_0_conv[0][0]
conv4_block1_3_bn (BatchNormali	(None, 19, 19, 1024) 4096	conv4_block1_3_conv[0][0]

conv4_block1_add (Add)	(None, 19, 19, 1024) 0	conv4_block1_0_bn[0][0] conv4_block1_3_bn[0][0]
conv4_block1_out (Activation)	(None, 19, 19, 1024) 0	conv4_block1_add[0][0]
conv4_block2_1_conv (Conv2D)	(None, 19, 19, 256) 262400	conv4_block1_out[0][0]
conv4_block2_1_bn (BatchNormali	(None, 19, 19, 256) 1024	conv4_block2_1_conv[0][0]
conv4_block2_1_relu (Activation	(None, 19, 19, 256) 0	conv4_block2_1_bn[0][0]
conv4_block2_2_conv (Conv2D)	(None, 19, 19, 256) 590080	conv4_block2_1_relu[0][0]
conv4_block2_2_bn (BatchNormali	(None, 19, 19, 256) 1024	conv4_block2_2_conv[0][0]
conv4_block2_2_relu (Activation	(None, 19, 19, 256) 0	conv4_block2_2_bn[0][0]
conv4_block2_3_conv (Conv2D)	(None, 19, 19, 1024) 263168	conv4_block2_2_relu[0][0]
conv4_block2_3_bn (BatchNormali	(None, 19, 19, 1024) 4096	conv4_block2_3_conv[0][0]
conv4_block2_add (Add)	(None, 19, 19, 1024) 0	conv4_block1_out[0][0] conv4_block2_3_bn[0][0]
conv4_block2_out (Activation)	(None, 19, 19, 1024) 0	conv4_block2_add[0][0]
conv4_block3_1_conv (Conv2D)	(None, 19, 19, 256) 262400	conv4_block2_out[0][0]

conv4_block3_1_bn (BatchNormali (None, 19, 19, 256) 1024 conv4_block3_1_conv[0][0]

conv4_block3_1_relu (Activation (None, 19, 19, 256) 0 conv4_block3_1_bn[0][0]

conv4_block3_2_conv (Conv2D) (None, 19, 19, 256) 590080 conv4_block3_1_relu[0][0]

conv4_block3_2_bn (BatchNormali (None, 19, 19, 256) 1024 conv4_block3_2_conv[0][0]

conv4_block3_2_relu (Activation (None, 19, 19, 256) 0 conv4_block3_2_bn[0][0]

conv4_block3_3_conv (Conv2D) (None, 19, 19, 1024) 263168 conv4_block3_2_relu[0][0]

conv4_block3_3_bn (BatchNormali (None, 19, 19, 1024) 4096 conv4_block3_3_conv[0][0]

conv4_block3_add (Add) (None, 19, 19, 1024) 0 conv4_block2_out[0][0]
conv4_block3_3_bn[0][0]

conv4_block3_out (Activation) (None, 19, 19, 1024) 0 conv4_block3_add[0][0]

conv4_block4_1_conv (Conv2D) (None, 19, 19, 256) 262400 conv4_block3_out[0][0]

conv4_block4_1_bn (BatchNormali (None, 19, 19, 256) 1024 conv4_block4_1_conv[0][0]

conv4_block4_1_relu (Activation (None, 19, 19, 256) 0 conv4_block4_1_bn[0][0]

conv4_block4_2_conv (Conv2D) (None, 19, 19, 256) 590080 conv4_block4_1_relu[0][0]

conv4_block4_2_bn (BatchNormali (None, 19, 19, 256) 1024 conv4_block4_2_conv[0][0]

conv4_block4_2_relu (Activation (None, 19, 19, 256) 0 conv4_block4_2_bn[0][0]

conv4_block4_3_conv (Conv2D) (None, 19, 19, 1024) 263168 conv4_block4_2_relu[0][0]

conv4_block4_3_bn (BatchNormali (None, 19, 19, 1024) 4096 conv4_block4_3_conv[0][0]

conv4_block4_add (Add) (None, 19, 19, 1024) 0 conv4_block3_out[0][0]
conv4_block4_3_bn[0][0]

conv4_block4_out (Activation) (None, 19, 19, 1024) 0 conv4_block4_add[0][0]

conv4_block5_1_conv (Conv2D) (None, 19, 19, 256) 262400 conv4_block4_out[0][0]

conv4_block5_1_bn (BatchNormali (None, 19, 19, 256) 1024 conv4_block5_1_conv[0][0]

conv4_block5_1_relu (Activation (None, 19, 19, 256) 0 conv4_block5_1_bn[0][0]

conv4_block5_2_conv (Conv2D) (None, 19, 19, 256) 590080 conv4_block5_1_relu[0][0]

conv4_block5_2_bn (BatchNormali (None, 19, 19, 256) 1024 conv4_block5_2_conv[0][0]

conv4_block5_2_relu (Activation (None, 19, 19, 256) 0 conv4_block5_2_bn[0][0]

conv4_block5_3_conv (Conv2D) (None, 19, 19, 1024) 263168 conv4_block5_2_relu[0][0]

conv4_block5_3_bn (BatchNormali (None, 19, 19, 1024) 4096 conv4_block5_3_conv[0][0]

conv4_block5_add (Add) (None, 19, 19, 1024) 0 conv4_block4_out[0][0]

conv4_block5_3_bn[0][0]

conv4_block5_out (Activation) (None, 19, 19, 1024) 0 conv4_block5_add[0][0]

conv4_block6_1_conv (Conv2D) (None, 19, 19, 256) 262400 conv4_block5_out[0][0]

conv4_block6_1_bn (BatchNormali (None, 19, 19, 256) 1024 conv4_block6_1_conv[0][0]

conv4_block6_1_relu (Activation (None, 19, 19, 256) 0 conv4_block6_1_bn[0][0]

conv4_block6_2_conv (Conv2D) (None, 19, 19, 256) 590080 conv4_block6_1_relu[0][0]

conv4_block6_2_bn (BatchNormali (None, 19, 19, 256) 1024 conv4_block6_2_conv[0][0]

conv4_block6_2_relu (Activation (None, 19, 19, 256) 0 conv4_block6_2_bn[0][0]

conv4_block6_3_conv (Conv2D) (None, 19, 19, 1024) 263168 conv4_block6_2_relu[0][0]

conv4_block6_3_bn (BatchNormali (None, 19, 19, 1024) 4096 conv4_block6_3_conv[0][0]

conv4_block6_add (Add) (None, 19, 19, 1024) 0 conv4_block5_out[0][0]

conv4_block6_3_bn[0][0]

conv4_block6_out (Activation) (None, 19, 19, 1024) 0 conv4_block6_add[0][0]

conv5_block1_1_conv (Conv2D) (None, 10, 10, 512) 524800 conv4_block6_out[0][0]

conv5_block1_1_bn (BatchNormali (None, 10, 10, 512) 2048 conv5_block1_1_conv[0][0]

conv5_block1_1_relu (Activation (None, 10, 10, 512) 0 conv5_block1_1_bn[0][0]

conv5_block1_2_conv (Conv2D) (None, 10, 10, 512) 2359808 conv5_block1_1_relu[0][0]

conv5_block1_2_bn (BatchNormali (None, 10, 10, 512) 2048 conv5_block1_2_conv[0][0]

conv5_block1_2_relu (Activation (None, 10, 10, 512) 0 conv5_block1_2_bn[0][0]

conv5_block1_0_conv (Conv2D) (None, 10, 10, 2048) 2099200 conv4_block6_out[0][0]

conv5_block1_3_conv (Conv2D) (None, 10, 10, 2048) 1050624 conv5_block1_2_relu[0][0]

conv5_block1_0_bn (BatchNormali (None, 10, 10, 2048) 8192 conv5_block1_0_conv[0][0]

conv5_block1_3_bn (BatchNormali (None, 10, 10, 2048) 8192 conv5_block1_3_conv[0][0]

conv5_block1_add (Add) (None, 10, 10, 2048) 0 conv5_block1_0_bn[0][0]
conv5_block1_3_bn[0][0]

conv5_block1_out (Activation) (None, 10, 10, 2048) 0 conv5_block1_add[0][0]

conv5_block2_1_conv (Conv2D) (None, 10, 10, 512) 1049088 conv5_block1_out[0][0]

conv5_block2_1_bn (BatchNormali (None, 10, 10, 512) 2048 conv5_block2_1_conv[0][0]

conv5_block2_1_relu (Activation (None, 10, 10, 512) 0 conv5_block2_1_bn[0][0]

conv5_block2_2_conv (Conv2D) (None, 10, 10, 512) 2359808 conv5_block2_1_relu[0][0]

conv5_block2_2_bn (BatchNormali (None, 10, 10, 512) 2048 conv5_block2_2_conv[0][0]

conv5_block2_2_relu (Activation (None, 10, 10, 512) 0 conv5_block2_2_bn[0][0]

conv5_block2_3_conv (Conv2D) (None, 10, 10, 2048) 1050624 conv5_block2_2_relu[0][0]

conv5_block2_3_bn (BatchNormali (None, 10, 10, 2048) 8192 conv5_block2_3_conv[0][0]

conv5_block2_add (Add) (None, 10, 10, 2048) 0 conv5_block1_out[0][0]
conv5_block2_3_bn[0][0]

conv5_block2_out (Activation) (None, 10, 10, 2048) 0 conv5_block2_add[0][0]

conv5_block3_1_conv (Conv2D) (None, 10, 10, 512) 1049088 conv5_block2_out[0][0]

conv5_block3_1_bn (BatchNormali (None, 10, 10, 512) 2048 conv5_block3_1_conv[0][0]

conv5_block3_1_relu (Activation (None, 10, 10, 512) 0 conv5_block3_1_bn[0][0]

conv5_block3_2_conv (Conv2D)	(None, 10, 10, 512)	2359808	conv5_block3_1_relu[0][0]
------------------------------	---------------------	---------	---------------------------

conv5_block3_2_bn (BatchNormali	(None, 10, 10, 512)	2048	conv5_block3_2_conv[0][0]
---------------------------------	---------------------	------	---------------------------

conv5_block3_2_relu (Activation	(None, 10, 10, 512)	0	conv5_block3_2_bn[0][0]
---------------------------------	---------------------	---	-------------------------

conv5_block3_3_conv (Conv2D)	(None, 10, 10, 2048)	1050624	conv5_block3_2_relu[0][0]
------------------------------	----------------------	---------	---------------------------

conv5_block3_3_bn (BatchNormali	(None, 10, 10, 2048)	8192	conv5_block3_3_conv[0][0]
---------------------------------	----------------------	------	---------------------------

conv5_block3_add (Add)	(None, 10, 10, 2048)	0	conv5_block2_out[0][0] conv5_block3_3_bn[0][0]
------------------------	----------------------	---	---

conv5_block3_out (Activation)	(None, 10, 10, 2048)	0	conv5_block3_add[0][0]
-------------------------------	----------------------	---	------------------------

flatten_1 (Flatten)	(None, 204800)	0	conv5_block3_out[0][0]
---------------------	----------------	---	------------------------

dense_4 (Dense)	(None, 1024)	209716224	flatten_1[0][0]
-----------------	--------------	-----------	-----------------

dropout_3 (Dropout)	(None, 1024)	0	dense_4[0][0]
---------------------	--------------	---	---------------

dense_5 (Dense)	(None, 512)	524800	dropout_3[0][0]
-----------------	-------------	--------	-----------------

dropout_4 (Dropout)	(None, 512)	0	dense_5[0][0]
---------------------	-------------	---	---------------

dense_6 (Dense)	(None, 256)	131328	dropout_4[0][0]
-----------------	-------------	--------	-----------------

dropout_5 (Dropout)	(None, 256)	0	dense_6[0][0]
---------------------	-------------	---	---------------

dense_7 (Dense)	(None, 2)	514	dropout_5[0][0]
-----------------	-----------	-----	-----------------

=====

=====

Total params: 233,960,578

Trainable params: 210,372,866

Non-trainable params: 23,587,712

7.PROGRAM DESIGN (MODULE & MODULE DESCRIPTION)

7.1 MODULES

There are four different modules required in the process. They are:

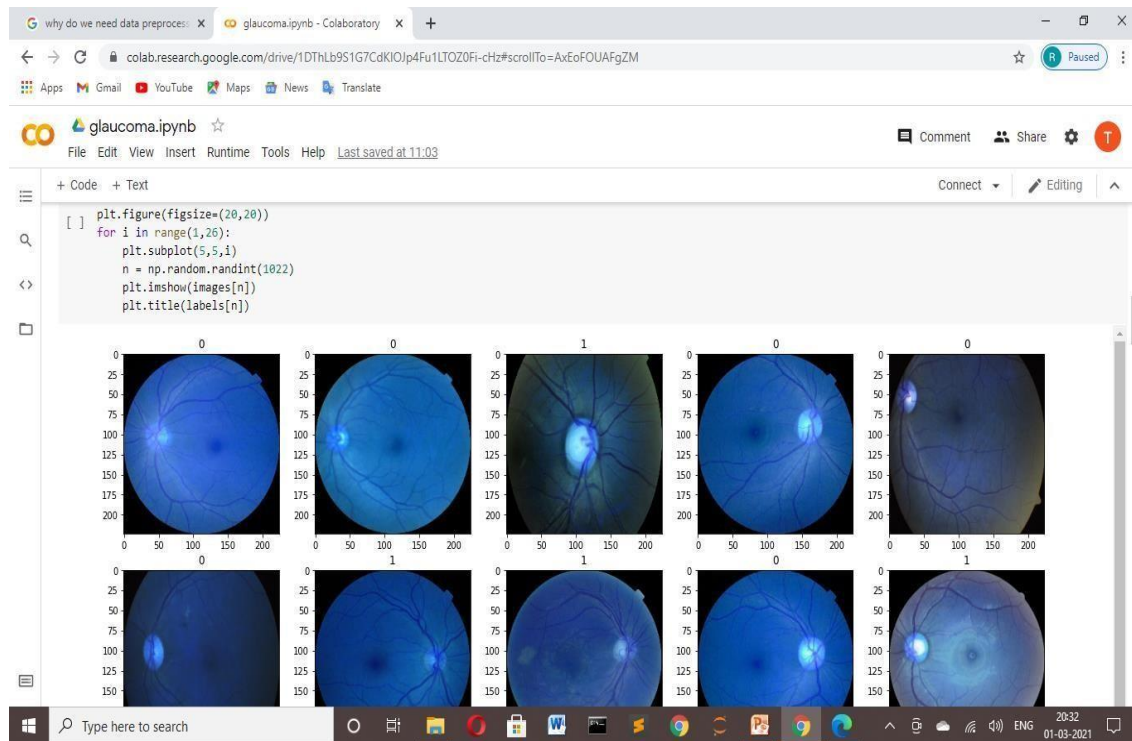
- Identification of required data
- Data Processing
- Definition of the training set
- Training the Machine Learning Model □ Test Evaluation

Identification of required data

- Data set is very important to determine effectiveness and accuracy of a system. A data set with large number of samples can effectively train and test a system and make it reliable to use.
- The dataset we use consists of 1022 images out of which 511 images are healthy eye images.

Data Processing

- Data preprocessing is an integral step in Machine Learning as the quality of data and the useful information that can be derived from it directly affects the ability of our model to learn.
- Therefore, it is extremely important that we preprocess our data before feeding it into our model.



Definition of the training set

- The training data is an initial set of data used to help a program understand how to apply technologies like neural networks to learn and produce sophisticated results.
- In this module, we allocate the proportion of training set for the machine learning model.

Training the Machine Learning Model

- The machine learning model is trained with the images in the training set.
- Every image is flipped and processed in multiple angles to train the model, to obtain more accuracy.

Test Evaluation

- Evaluation is a process during development of the model to check whether the model is best fit for the given problem and corresponding data.
- Keras model provides a function, evaluate which does the evaluation of the model.

8.TESTING

Input:

```
history = resnet50_model.fit_generator(generator = train_generator, epochs = NUM_EPOCHS,  
steps_per_epoch = 100, shuffle = True, validation_data = test_generator)
```

output:

Epoch 1/100

65/100 [=====>] - 67s 1s/step - loss: 1.8080 - accuracy: 0.6288 -
val_loss: 1.3013 - val_accuracy: 0.7385

Input:

```
import matplotlib.pyplot as plt
```

```
%matplotlib inline
```

```
image_batch,label_batch = train_generator.next()
```

```
print(len(image_batch)) for i in
```

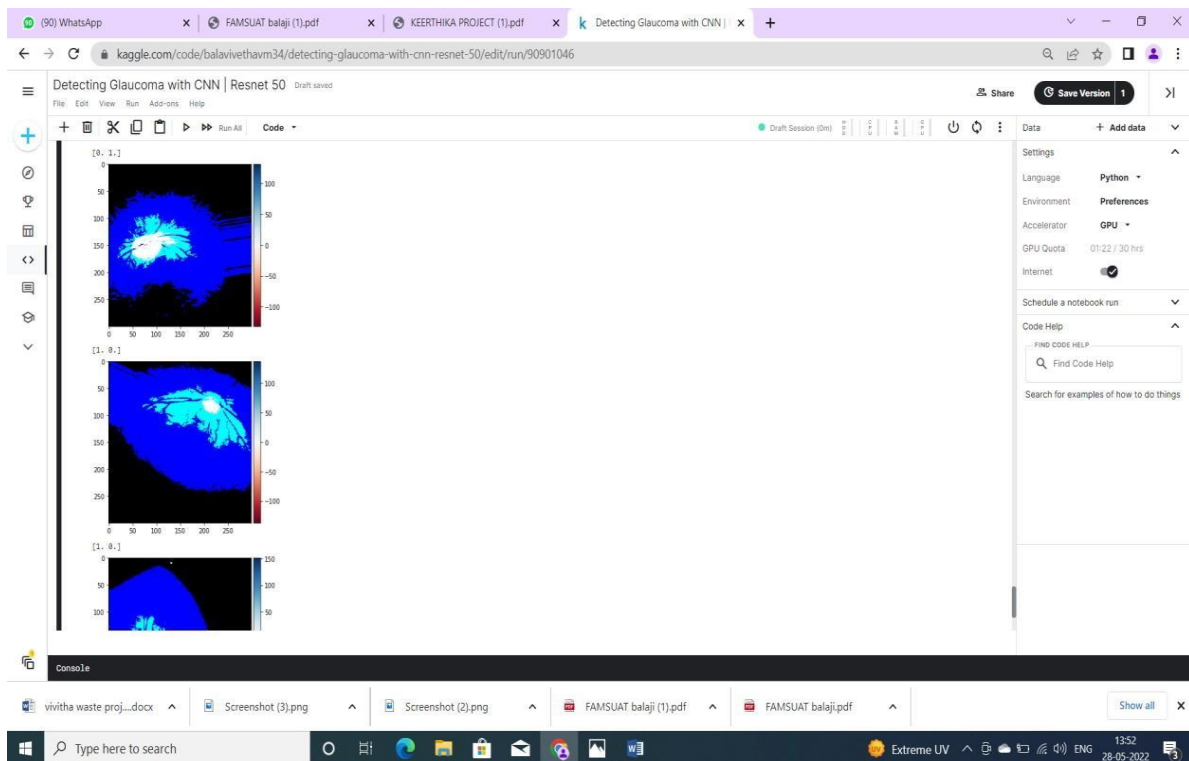
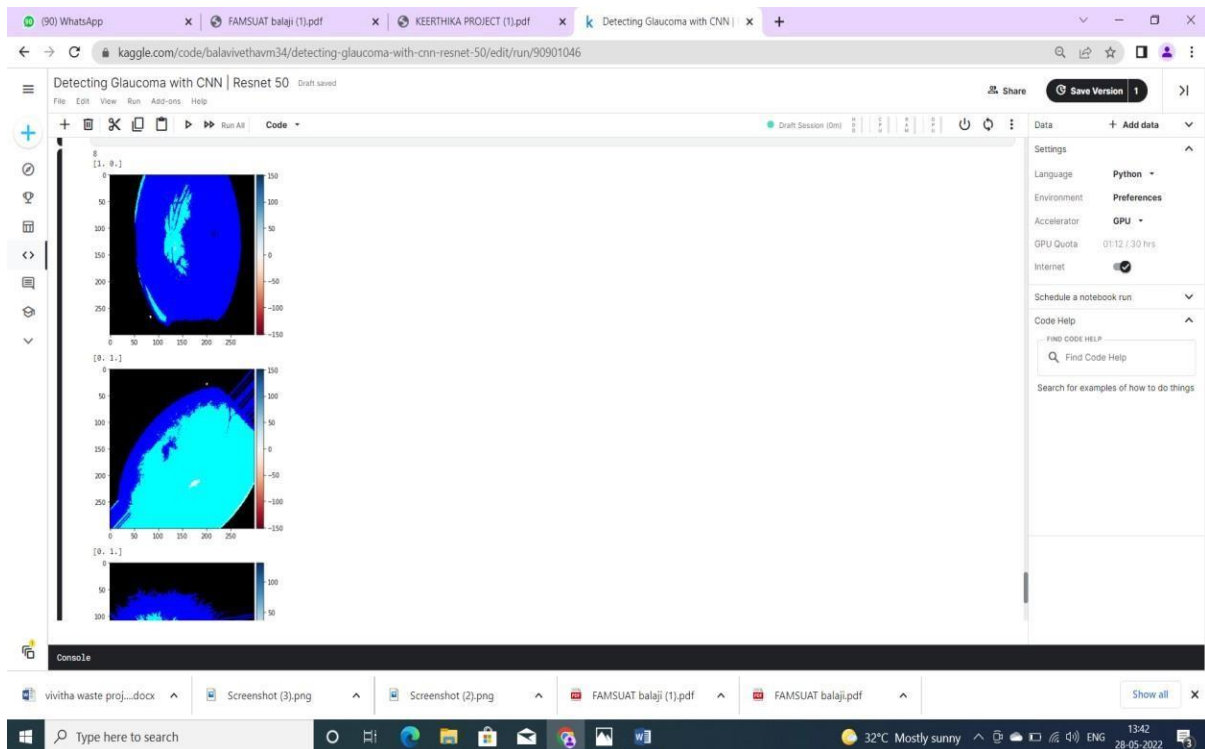
```
range(0,len(image_batch)):
```

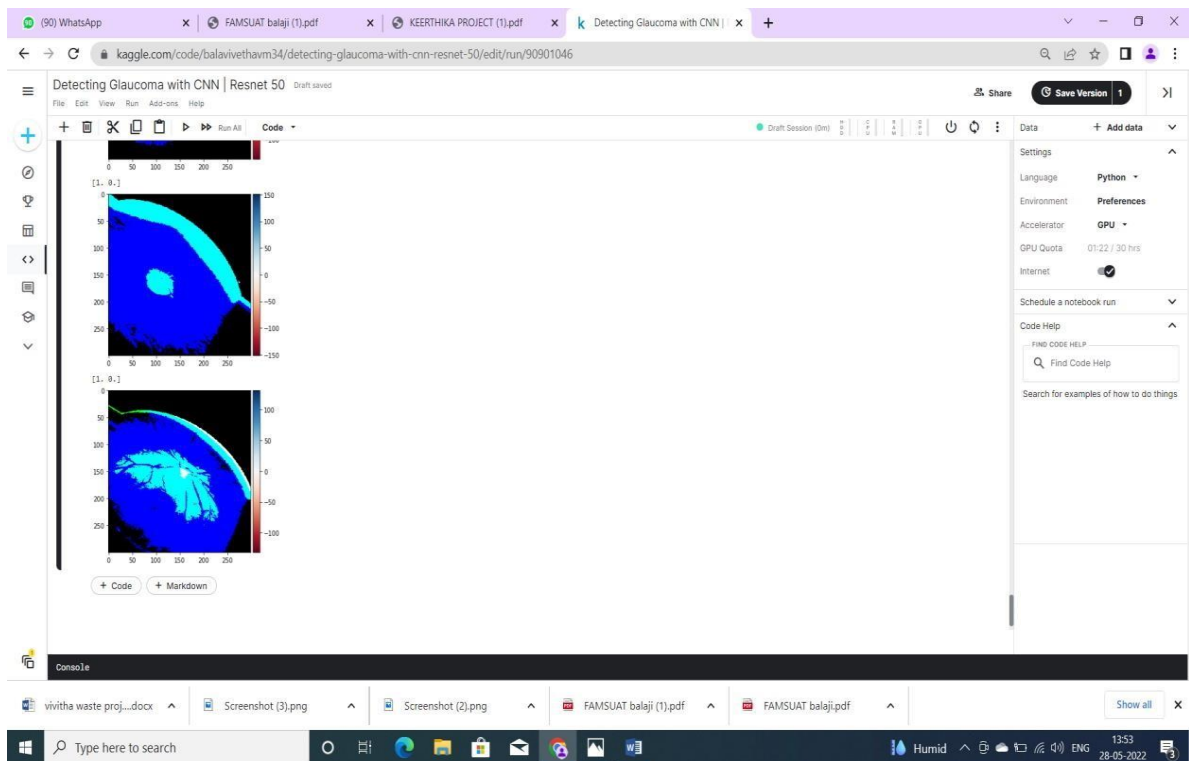
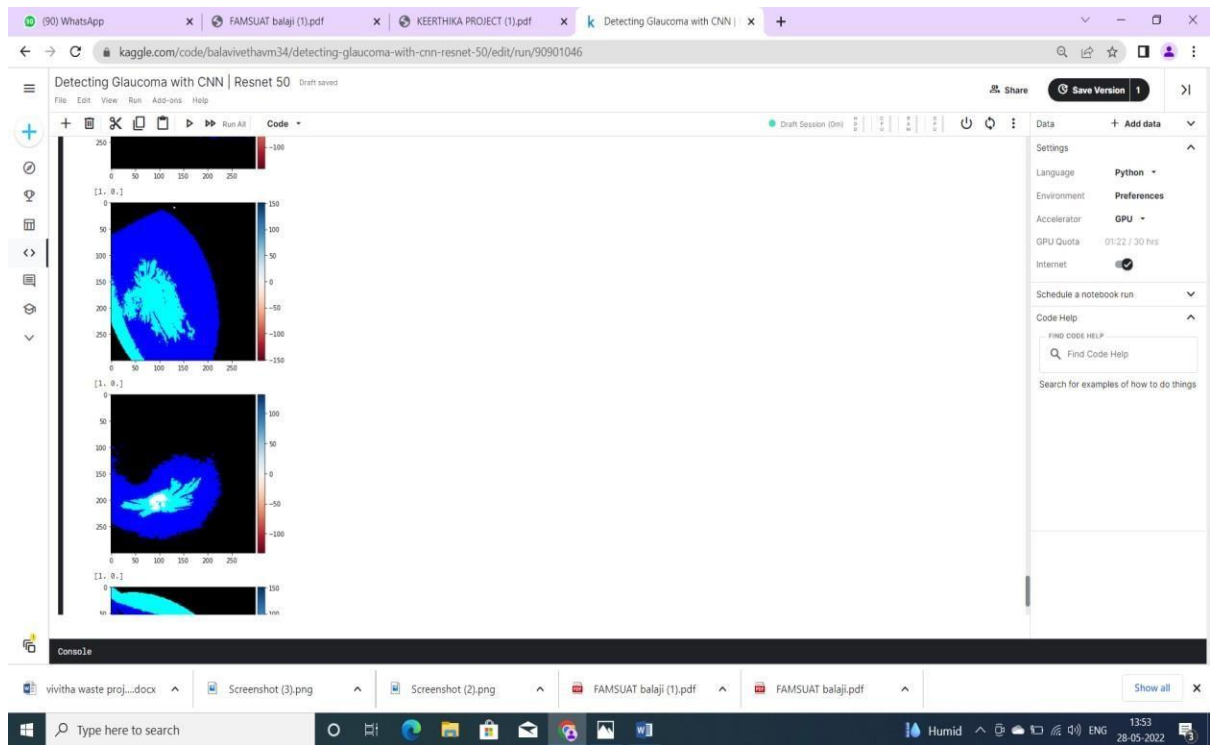
```
    image = image_batch[i]
```

```
    print(label_batch[i])
```

```
    imshow(image)
```

Output:





9.CONCLUSION

Glaucoma is one of the primary concerns for the human eye. The vision of the eye can be lost; the glaucoma symptoms are seen in the human eye. The glaucoma is a significant defect that may result in blindness if pre-stage detection is not done. So to detect this disease, a mechanism that uses the deep learning CNN is proposed for the analysis. The proposed mechanism is based on the six-layer architecture, and CNN will classify the patterns observed for the glaucoma detection in the patient's eye images. Two datasets are used here; they are ORIGA and the SCES dataset. The proposed method worked very well and the results obtained are satisfactory. The AUC values are carried out from our experiment. The values are obtained for both the datasets and compared with the existing state of the art algorithm. The earned value is .882 and .822 for the SCES and the ORIGA dataset respectively. The proposed method can be used for glaucoma detection.

9.1 FUTURE ENHANCEMENT

- Use reinforcement machine learning technique for training the model.
- Implement existing system as a mobile application.

10.APPENDIX

A. SOURCE CODE

```
import os      import cv2
import numpy as np
from tqdm import tqdm
import matplotlib.pyplot as plt
import pandas as pd

def crop_image_from_gray(img,tol=7):

    if img.ndim==2:
        mask =img>tol
        return img[np.ix_(mask.any(1),mask.any(0))]

    elif img.ndim==3:
        gray_img= cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
        mask =gray_img>tol
        check_shape=img[:, :,0][np.ix_(mask.any(1),mask.any(0))].shape[0]

        if (check_shape==0):
            return img

        else:
            img1=img[:, :,0][np.ix_(mask.any(1),mask.any(0))]
            img2=img[:, :,1][np.ix_(mask.any(1),mask.any(0))]
            img3=img[:, :,2][np.ix_(mask.any(1),mask.any(0))]
```

```
img=np.stack([img1,img2,img3],axis=-1)
```

```
returnimg
```

```
path = ['./glaucomadataset/Non Glaucoma',  
        './glaucomadataset/Glaucoma'] images = [] labels = []
```

```
forn,iinenumerate(path): for
```

```
j in tqdm(os.listdir(i)):
```

```
    img_path=os.path.join(i,j)      img=  
    cv2.imread(img_path)  
    img=crop_image_from_gray(img,tol=7  
)  img= cv2.resize(img, (224,224))  
    images.append(img) labels.append(n)
```

```
images =np.array(images)/255 labels  
=np.array(labels)
```

```
plt.figure(figsize=(20,20))
```

```
foriinrange(1,26):
```

```
    plt.subplot(5,5,i)
```

```
    n =np.random.randint(1022)
```

```
    plt.imshow(images[n]) plt.title(labels[n])
```

```
fromsklearn.utilsimport shuffle
```

```
fromsklearn.model_selectionimporttrain_test_split
```

```

images, labels = shuffle(images, labels, random_state=32) x_train, x_valid, y_train,
y_valid=train_test_split(images, labels, test_size=0.15, random_state=4
4)
x_train, x_test, y_train, y_test=train_test_split(x_train, y_train, test_size=0.15, random_state=40)

fromkeras.preprocessing.imageimportImageDataGenerator
datagen=ImageDataGenerator(zoom_range=0.2, horizontal_flip=True,
vertical_flip=True,fill_mode='constant', cval=0.)
train_gen=datagen.flow(x_train, y_train, batch_size=32)

fromkeras.modelsimport Sequential, model_from_json fromkeras.layersimport Conv2D, Dense,
Dropout, BatchNormalization, Flatten, MaxPool2D, GlobalAveragePooling2D
fromkeras.callbacksimportModelCheckpoint, ReduceLROnPlateau
fromkeras.applications.densenetimport DenseNet121 tr = DenseNet121(weights='imagenet',
include_top=False, input_shape=(224,224,3))

model = Sequential() model.add(tr)
model.add(GlobalAveragePooling2D()) model.add(Dropout(0.5))
model.add(Dense(1, activation='sigmoid')) model.compile(loss='binary_crossentropy',
optimizer='adam', metrics=['accuracy']) ck =ModelCheckpoint('glaucoma_weights.hdf5',
monitor='val_loss', save_best_only=True, mode='auto', verbose=1) re
=ReduceLROnPlateau(monitor='val_loss', mode='auto', factor=0.2, patience=4, verbose=1)
model.summary()

history =model.fit_generator(train_gen, epochs=12,
verbose=1,
validation_data=(x_valid,y_valid),
callbacks=[ck,re])

history_df=pd.DataFrame(history.history)
history_df[['loss', 'val_loss']].plot()
history_df[['accuracy',
'val_accuracy']].plot()

```

```

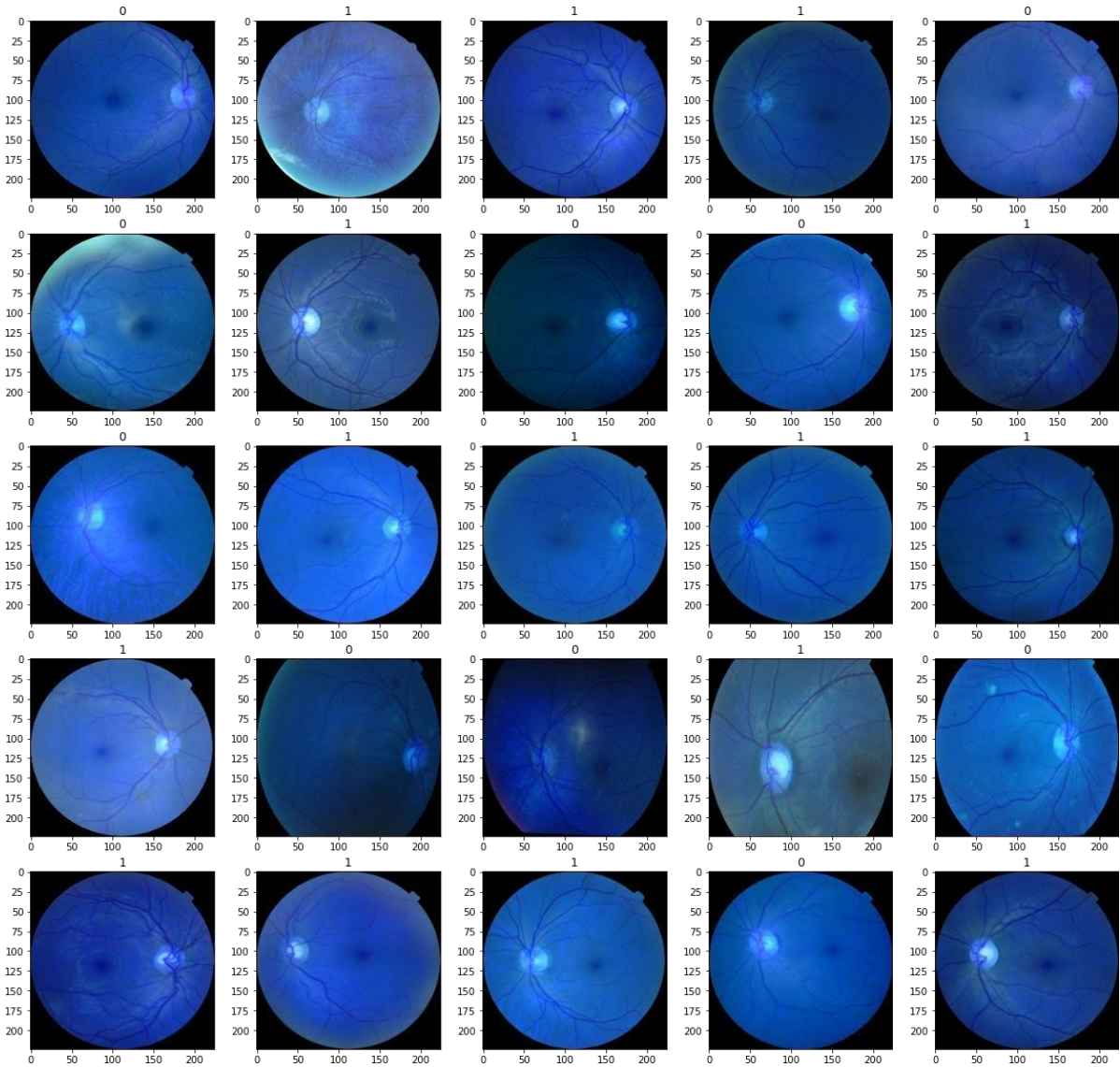
pred=model.evaluate(x_test, y_test)
print('Test Accuracy:', pred[1]*100)
print('Test Loss:', pred[0]*100)
# AUC Curvw

fromsklearn.metricsimportroc_curve
y_pred_keras=model.predict(x_test).ravel() fpr_keras, tpr_keras,
thresholds_keras=roc_curve(y_test, y_pred_keras)
fromsklearn.metricsimportauc auc_keras=auc(fpr_keras, tpr_keras)
print(auc_keras*100)

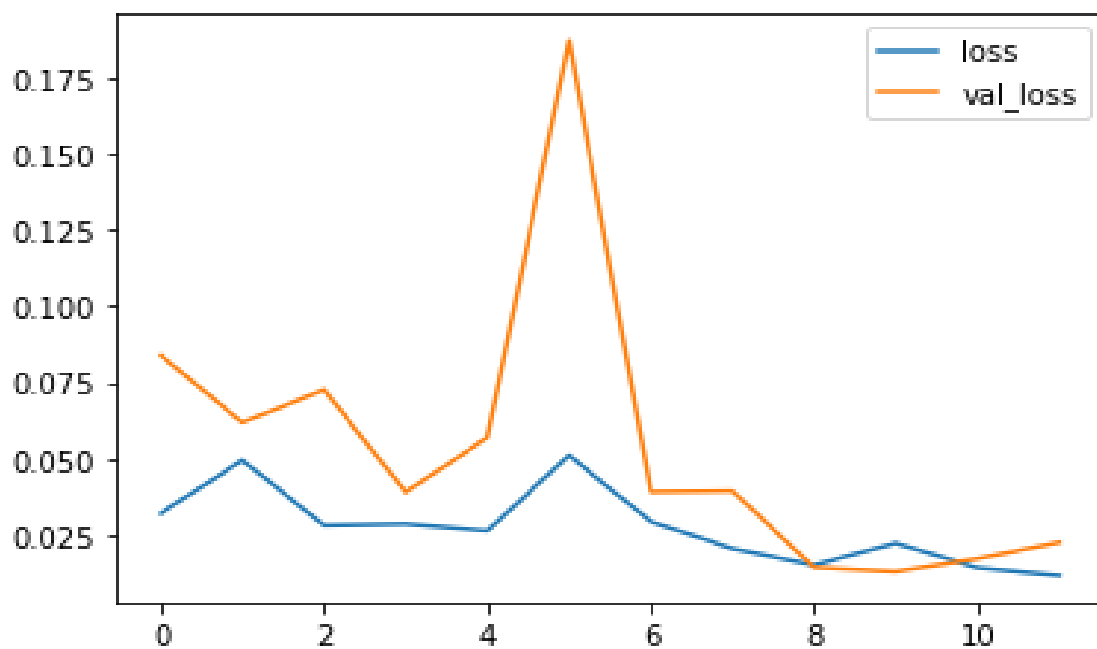
plt.figure(1) plt.plot([0, 1], [0, 1], 'k--') plt.plot(fpr_keras, tpr_keras,
label='Keras (area = {:.3f})'.format(auc_keras)) plt.xlabel('False positive
rate') plt.ylabel('True positive rate') plt.title('ROC curve')
plt.legend(loc='best') plt.show()

```

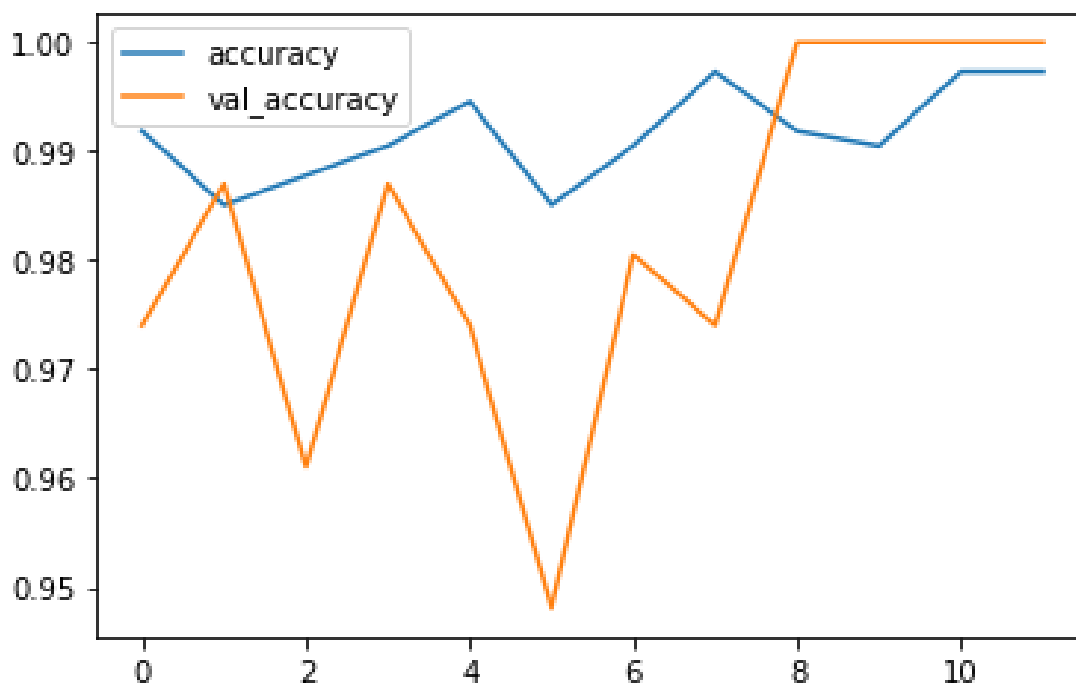
B. SAMPLE INPUT AND OUPUT



B.1 Glaucoma Display



B.1 Loss output



B.1 Accuracy output

11.REFERENCE

1. T. Khalil, S. Khalid, A. M. Syed, "Review of Machine Learning Techniques for Glaucoma Detection and Prediction", Science and Information Conference 2014, London, UK,438
2. R.Panda, N. B. Puhan, A.Rao, D.Padhyand,G.Panda, "Recurrent Neural Network Based Retinal Nerve Fibre Layer Defect Detection in Early Glaucoma", School of Electrical Sciences, IIT Bhubaneswar, India Glaucoma Diagnostic Services, L.V. Prasad Eye Institute Bhubaneswar, India
3. K. Choudhary, P. Maheshwari and S. Wadhwa, "Glaucoma Detection using Cross-Validation Algorithm: A comparative evaluation on Rapid miner", 978-1-4799-4562-7/14/\$31.00©2014IEEE
4. S. J. Kim, K. J. Cho and S. Oh, "Development of machine learning models for diagnosis of glaucoma", <https://doi.org/10.1371/journal.pone.0177726>May23,2017
5. S. C. Shetty and P. Gutte," A Novel Approach for Glaucoma Detection Using Fractal Analysis", 978-1-5386-3624- 4/18/\$31.00c2018IEEE
6. L. Li, M. Xu, H. Liu, Y. Li, X. Wang, L. Jiang, Z. Wang, X. Fan, and N. Wang, "A Large-scale Database and a CNN Model for Attention-based Glaucoma Detection", IEEE TransactionsOnMedicalImaging,2019,1-11
7. M. Ahn, S. Kim, K.S. Ahn, S.H. Cho, K. B. Lee, U.S. Kim, "A deep learning model for detection of both advanced & early glaucoma using fundus photography", <https://doi.org/10.1371/journal.pone.0207982> November 27, 2018
8. Li, J. Cheng, D. W. K. Wong and J. Liu," Integrating Holistic and Local Deep Features for Glaucoma Classification", 978-1-4577-0220-4/16/2016IEEE

9. Serener and S. Serte, “Transfer Learning for Early and Advanced Glaucoma Detection with Convolutional Neural Networks”, 978-1-7281-2420-9/19/ 2019IEEE

10. R. Daneshvar, A. Yarmohammadi, R. Alizadeh, S. Henry, S. Law, J. Caproli, and K. Mahdavi, “Prediction of Glaucoma Progression with Structural Parameters: Comparison of Optical Coherence Tomography and Clinical Disc Parameters”, American Journal of Ophthalmology, December 2019

11. Guangzhou, K. Omodaka, K. Hashimoto, S. Tsuda, Y. Shiga, N. Takada, T. Kikawa, H. Yokota, M. Akiba and T. Nakazawa, “ Glaucoma Diagnosis with Machine Learning Based on Optical Coherence Tomography & Color Fundus Images”, Journal of Healthcare Engineering Volume 2019,

12. J. Carrillo, L. Bautista, J. Villamizar, J. Rueda, M. Sanchez, and D. Rueda, “Glaucoma Detection Using Fundus Images of the Eye,” *2019 22nd Symp. Image, Signal Process. Artif. Vision, STSIVA 2019 - Conf. Proc.*, pp. 1–4, 2019, DOI: 10.1109/STSIVA.2019.8730250.

13. T. Khalil, M. U. Akram, S. Khalid, and A. Jameel, “ An overview of automated glaucoma detection,” *Proc. Comput. Conf. 2017*, vol. 2018-January, no. July, pp. 620–632, 2018, DOI: 10.1109/SAI.2017.8252161.

14. Sengar, M. K. Dutta, R. Burget, and M. Ranjoha, “Automated detection of suspected glaucoma in digital fundus images,” *2017 40th Int. Conf. Telecommun. Signal Process. TSP, 2017*, vol. 2017-January, pp. 749–752, 2017, DOI: 10.1109/TSP.2017.8076088.

15. A.Bashar, "Survey on evolving deep learning neural network architectures", Journal of Artificial Intelligence and Capsule Networks (2019) Vol.01/ Issue.02 Pages: 73-82 <http://irojournals.com/aicn/> DOI: <https://doi.org/10.36548/jaicn.2019.2.003>.
16. T. Vijayakumar, COMPARATIVE STUDY OF CAPSULE NEURAL NETWORK IN VARIOUS APPLICATIONS, Journal of Artificial Intelligence and Capsule Networks (2019) Vol.01/ No. 01
Pages: - 27 <http://irojournals.com/aicn/> DOI: <https://doi.org/10.36548/jaicn.2019.1.003>

