# GLAUCOMA DETECTION USING CONVOLUTIONAL NEURAL NETWORK

**A PROJECT REPORT**

***Submitted by***

**NISHANT BHUSHAN C [211417104168]**

***in partial fulfillment for the award of the degree***

***of***

**BACHELOR OF ENGINEERING**

***in***

**COMPUTER SCIENCE AND ENGINEERING**

**PANIMALAR ENGINEERING COLLEGE**

**ANNA UNIVERSITY: CHENNAI 600 025**

**APRIL 2021**

## ANNA UNIVERSITY : CHENNAI 600 025

## BONAFIDE CERTIFICATE

Certified that this project report **“GLAUCOMA DETECTION USING CONVOLUTIONAL NEURAL NETWORK”** is the bonafide work of “**NISHANT BHUSHAN C (211417104168)”** who carried out the project work under my supervision.

| | |
|---|---|
| **SIGNATURE** | **SIGNATURE** |
| **Dr. S. MURUGAVALLI, M.E., Ph.D.,** | **Mr. S.A.K. JAINULABUDEEN, M.Tech.,** |
| **PROFESSOR** | **SUPERVISOR** |
| **HEAD OF THE DEPARTMENT** | **ASSISTANT PROFESSOR** |
| DEPARTMENT OF CSE, | DEPARTMENT OF CSE, |
| PANIMALAR ENGINEERING COLLEGE, | PANIMALAR ENGINEERING COLLEGE, |
| NASARATHPETTAI, | NASARATHPETTAI, |
| POONAMALLEE, | POONAMALLEE, |
| CHENNAI-600 123. | CHENNAI-600 123. |

Certified that the above candidate was examined in the Anna University Project Viva-

Voce Examination held on.........................

| | |
|---|---|
| **INTERNAL EXAMINER** | **EXTERNAL EXAMINER** |

# ACKNOWLEDGEMENT

My sincere, humble and heartfelt thanks and praises to our honourable Founder and Chairman Late **Dr. JEPPIAAR, M.A., B.L., Ph.D.,** for his sincere Endeavor in educating us in his reputed institution.

We would like to express our deep gratitude to our respected Secretary and Correspondent **Dr. P. CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We express our sincere thanks to our Directors **TMT. C. VIJAYARAJESWARI**, **THIRU. C. SAKTHI KUMAR, M.E., Ph.D.,** and **TMT. SARANYA SREE SAKTHI KUMAR, B.E.,** for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our Principal **Dr. K. MANI, M.E., Ph.D.** for his timely concern and encouragement provided to us throughout the course.

We thank the Head of the CSE Department, **Dr. S. MURUGAVALLI , M.E., Ph.D.,** for the support extended throughout the project.

We would like to thank my **Supervisor S.A.K. JAINULABUDEEN, M.Tech.,** and all the faculty members of the Department of CSE for their advice and suggestions for the successful completion of the project.

**NISHANT BHUSHAN C**

## ABSTRACT

Glaucoma is often linked to a build-up of pressure inside the eyes. Glaucoma tends to run in families and one usually doesn’t get it until later in life. The increased pressure in eyes, called intraocular pressure, can damage the optic nerve, which sends images to the brain. If the damage worsens, glaucoma can cause permanent vision loss or even total blindness within a few years. Most people with glaucoma have no early symptoms or pain. One must visit the eye doctor regularly so they can diagnose and treat glaucoma before one has long-term vision loss. If a person loses his vision, it can’t be brought back. But, lowering eye pressure can help keep the sight that he has. Most people with glaucoma who follow their treatment plan and have regular eye exams are able to keep their vision. Glaucoma is a chronic and irreversible eye disease, which leads to deterioration in vision and quality of life. we develop a Deep Learning (DL) with convolutional neural network for automated glaucoma diagnosis. Deep learning systems, such as convolutional neural networks (CNNs), can infer a hierarchical representation of images to discriminate between glaucoma and non-glaucoma patterns for diagnostic decisions. The model is trained with the ROI of RIGA, DRISHTI-GS1 dataset. The Network architecture used gives great accuracy. A graphical user interface is used to diagnose the condition of test images and give a graphical analysis of the patients.

# TABLE OF CONTENTS

## LIST OF FIGURES

# LIST OF ABBREVATIONS

| ABBREVIATION | DESCRIPTION |
|---|---|
| IOP | Intra Ocular Pressure |
| CDR | Call Detailed Record |
| SGD | Stochastic Gradient Descent |
| CNN | Convolutional Neural Network |
| DE-CNN | Deconvolution Neural Network |
| CLARANS | Clustering Large Application Based Upon Randomized Search |
| GMM | Gaussian Mixture Model |

# CHAPTER 1

# INTRODUCTION

Glaucoma is one of the most hazardous ocular diseases. It is a set of disorders, categorized mostly by high intraocular pressure (IOP) causing damage to the optic nerve. According to the World Health Organization it is second foremost cause of blindness worldwide. Glaucoma is also known as snitch thief of sight because resembling a hushed slayer it usually has no indicators till permanent vision loss occurs. Glaucoma is often linked to a build-up of pressure inside the eyes. Glaucoma tends to run in families and one usually doesn't get it until later in life. The increased pressure in eyes, called intraocular pressure, can damage the optic nerve, which sends images to the brain. If the damage worsens, glaucoma can cause permanent vision loss or even total blindness within a few years. Most people with glaucoma have no early symptoms or pain. One must visit the eye doctor regularly so they can diagnose and treat glaucoma before one has long-term vision loss. If a person loses his vision, it can't be brought back. But, lowering eye pressure can help keep the sight that he has. Most people with glaucoma who follow their treatment plan and have regular eye exams are able to keep their vision.

# CHAPTER 2

# LITERATURE SURVEY

**[1]"Detection of Glaucoma Using Retinal Fundus Images" Hafsah Ahmad, Abubakar Yamin, Aqsa Shakeel, Syed Omer Gillani, Umar Ansari(IEEE 2014)** .

This paper proposes an image processing technique for the detection of glaucoma which mainly affects the optic disc by increasing the cup size. During early stages it was difficult to detect Glaucoma, which is in fact second leading cause of blindness. In this paper glaucoma is categorized through extraction of features from retinal fundus images. The features include (i) Cup to Disc Ratio (CDR), which is one of the primary physiological parameters for the diagnosis of glaucoma and (ii) Ratio of Neuroretinal Rim in inferior, superior, temporal and nasal quadrants i.e. (ISNT quadrants) for verification of the IS NT rule. The novel technique is implemented on 80 retinal images and an accuracy of 97.5% is achieved taking an average computational time of 0.8141 seconds.

**[2]"Automatic Glaucoma Detection by Using Funduscopic Images", Atheesan S., Yashothara S.(IEEE 2016).**

Glaucoma is a group of related eye disorders that cause damage to the optic nerve that carries information from the eye to the brain which can get worse over time and lead to blindness. It is very important that glaucoma is detected as early as possible for proper treatment. In this paper, we have proposed a Convolutional Neural Network (CNN) system for early detection of Glaucoma. Initially, eye images are augmented to generate data for Deep learning. The eye images are then pre- processed to remove noise using Gaussian Blur technique and make the image suitable for further processing. The system is trained using the pre-processed images and when new input images are given to the system it classifies them as normal eye or glaucoma eye based on the features extracted during training.

**[3]”Automated Detection of Suspected Glaucoma in Digital Fundus Images”, Namita Sengar, Malay Kishore Dutta, Radim Burget, Martin Ranjoha(IEEE 2017)**

Glaucoma is the Second leading causes of blindness across the world, it is an eye disease that affects the retina and optic nerve of eye which carries signals to the eye. This disease can lead to permanent blindness if not treated at earlier stage. The increased intraocular pressure which is main cause of glaucoma damages the optic nerve which sends images to the brain, the diameter of the optic cup within optic disc region is increased due to this increased retinal pressure. The increased cup to disc ratio (CDR) results in the loss of optic nerve fibres that are connected to the retina by disc area. CDR is the important structural features for differentiating between healthy and a glaucomatous eye. The objective of this paper is to present review of methods for automated detection of Glaucoma from fundus images of eye that assist in the progressive development of computer aided systems.

**[4] Budai A, Bock R, Maier A, Hornegger J, Michelson G. Robust vessel segmentation in fundus images. Int J Biomed Imag. 2013.**

One of the most common modalities to examine the human eye is the eye-fundus photograph. The evaluation of fundus photographs is carried out by medical experts during time-consuming visual inspection. Our aim is to accelerate this process using computer aided diagnosis. As a first step, it is necessary to segment structures in the images for tissue differentiation. As the eye is the only organ, where the vasculature can be imaged in an in vivo and noninterventional way without using expensive scanners, the vessel tree is one of the most interesting and important structures to analyse. The quality and resolution of fundus images are rapidly increasing. Thus, segmentation methods need to be adapted to the new challenges of high resolutions. In this paper, we present a method to reduce calculation time, achieve high accuracy, and increase sensitivity compared to the original Frangi method. This method contains approaches to avoid potential problems like specular reflexes of thick vessels. The proposed method is evaluated using the STARE and DRIVE databases and we propose a new high resolution fundus database to compare it to the state-of-the-art algorithms. The results show an average accuracy above 94% and low computational needs. This outperforms state-of-the-art methods.

**[5] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. 2014. ArXiv e-prints arxiv:abs/1409.1556**

In this work we investigate the effect of the convolutional network depth on its accuracy in the large-scale image recognition setting. Our main contribution is a thorough evaluation of networks of increasing depth using an architecture with very small ( 3 × 3) convolution filters, which shows that a significant improvement on the prior-art configurations can be achieved by pushing the depth to 16–19 weight layers. These findings were the basis of our ImageNet Challenge 2014 submission, where our team secured the first and the second places in the localisation and classification tracks respectively. We also show that our representations generalise well to other datasets, where they achieve state-of-the-art results. We have made our two best-performing ConvNet models publicly available to facilitate further research on the use of deep visual representations in computer vision.

**[6] "A Comprehensive Retinal Image Dataset for the Assessment of Glaucoma from the Optic Nerve Head Analysis" Jayanthi Sivaswamy, S.R.Krishnadas, Arunava Chakravarty, Gopal Datt Joshi, Ujjwal1 and Tabish Abbas Syed.**

In this work we investigated on All of the photographs were taken with the cooperation of visitors to Aravind Eye Hospital in Madurai. Glaucoma is a disease that affects the eyes. Clinical investigators chose patients based on a variety of factors. Observations made during the examination Patients between the ages of 40 and 60 were chosen. Males and females are about equal in age at 80 years old. Patients who get a regular refraction test and are not found to be deficient, we have glaucoma. All photos were collected with the eyes dilated using the following data collection protocol: centered on OD with a Field of View of 30-degrees and resolution of 2896 x 1944 pixels; The image format PNG is uncompressed. There are no other options. The acquisition procedure was subjected to imaging limitations. Four glaucoma patients supplied ground truth for each image. professionals with more than three years' experience. Poorquality photos were deleted due to poor contrast, OD region location, and other factors. Fundus area (image region) is included in this dataset. The original has been retrieved (containing retinal structures). By removing the non-fundus mask zone surrounding the image a 2047 x 1760-pixel picture.

# CHAPTER 3

# SYSTEM ANALYSIS

## 3.1 EXISTING SYSTEM

Traditional methods of detecting glaucoma include an eye doctor analyzing the images and finding the abnormalities in it. This method is very time consuming and not always accurate because the image contains noise and other factors which make it difficult for proper analysis. Also, if a machine trained for analysis, it becomes more accurate than human analysis.

## 3.2 PROPOSED SYSTEM

The system proposed a novel semi-automatic detection method based on CNN deep Learning Techniques. There are many computer-based techniques are present to identify Glaucoma disease, but these systems have advantages as well as disadvantages. We have developed a system having more advantages and fewer drawbacks by using a smarter algorithm. The aim of our project is to improve the performance and accuracy of the system by using a different algorithm.

## 3.3 HARWARE REQUIREMENTS

| | |
|---|---|
| Processor | : Intel Pentium Dual Core 2.00GHz |
| Hard disk | : 500 GB |
| RAM | : 8 GB (minimum) |

## 3.4 SOFTWARE REQUIREMENTS

- Python 3.6.4 Version

## 3.5 SOFTWARE SPECIFICATION

### 3.5.1 Machine learning

**Machine learning** (**ML**) is the study of computer algorithms that improve automatically through experience and by the use of data. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.

The types of machine learning algorithms are mainly divided into four categories:

- **Supervised learning,**
- **Un-supervised learning,**
- **Semi-supervised learning,**
- **Reinforcement learning**.

❖ **Supervised learning**

Supervised learning algorithms build a mathematical model of a set of data that contains both the inputs and the desired outputs. The data is known as training data, and consists of a set of training examples. Each training example has one or more inputs and the desired output, also known as a supervisory signal. In the mathematical model, each training example is represented by an array or vector, sometimes called a feature vector, and the training data is represented by a matrix. Through iterative optimization of an objective function, supervised learning algorithms learn a function that can be used to predict the output associated with new inputs. An optimal function will allow the algorithm to correctly determine the output for inputs that were not a part of the training data. An algorithm that improves the accuracy of its outputs or predictions over time is said to have learned to perform that task.

## CLASSIFICATION

As the name suggests, Classification is the task of "classifying things" into sub- categories. But, by a machine. If that doesn't sound like much, imagine your computer being able to differentiate between you and a stranger. Between a potato and a tomato. Between an A grade and a F. In Machine Learning and Statistics, Classification is the problem of identifying to which of a set of categories (sub populations), a new observation belongs to, on the basis of a training set of data containing observations and whose categories membership is known.

## TYPES OF CLASSIFICATION

Classification is of two types:

- Binary Classification
- Multiclass Classification

### Binary Classification

When we have to categorize given data into 2 distinct classes. Example – On the basis of given health conditions of a person, we have to determine whether the person has a certain disease or not.

### Multiclass Classification

The number of classes is more than 2. For Example

– On the basis of data about different species of flowers, we have to determine which specie does our observation belong to

Fig 2 : Binary and Multiclass Classification. Here x1 and x2 are our variables upon which the class is predicted.Suppose we have to predict whether a given patient has a certain disease or not, on the basis of 3 variables, called features. Which means there are two possible outcomes:

1. The patient has the said disease. Basically a result labelled "Yes" or "True".

2. The patient is disease free. A result labelled "No" or "False".

This is a binary classification problem. We have a set of observations called training data set, which comprises of sample data with actual classification results. We train a model, called Classifier on this data set, and use that model to predict whether a certain patient will have the

1. X : pre-classified data, in the form of a N*M matrix. N is the no. of observations and M is the number of features

2. y : An N-d vector corresponding to predicted classes for each of the N observations.

3. Feature Extraction : Extracting valuable information from input X using a series of transforms.

4. ML Model : The "Classifier" we'll train.

5. y' : Labels predicted by the Classifier.

6. Quality Metric : Metric used for measuring the performance of the model.

7. ML Algorithm : The algorithm that is used to update weights w', which update the model and "learns" iteratively.

Types of Classifiers (Algorithms)

There are various types of classifiers. Some of them are :

- Linear Classifiers : Logistic Regression
- Tree Based Classifiers : Decision Tree Classifier
- Support Vector Machines
- Artificial Neural Networks
- Bayesian Regression
- Gaussian Naive Bayes Classifiers
- Stochastic Gradient Descent (SGD) Classifier
- Ensemble Methods : Random Forests, AdaBoost, Bagging Classifier, Voting Classifier, ExtraTrees Classifier

Practical Applications of Classification

- Google’s self driving car uses deep learning enabled classification techniques which enables it to detect and classify obstacles.
- Spam E-mail filtering is one of the most widespread and well recognized uses of Classification techniques.

- Detecting Health Problems, Facial Recognition, Speech Recognition, Object Detection, Sentiment Analysis all use Classification at their core.

**REGRESSION**

A regression problem is when the output variable is a real or continuous value, such as “salary” or “weight”. Many different models can be used, the simplest is the linear regression. It tries to fit data with the best hyper-plane which goes through the points.

- **Un-supervised learning**

Un-supervised learning algorithms take a set of data that contains only inputs, and find structure in the data, like grouping or clustering of data points. The algorithms, therefore, learn from test data that has not been labeled, classified or categorized. Instead of responding to feedback, unsupervised learning algorithms identify commonalities in the data and react based on the presence or absence of such commonalities in each new piece of data. A central application of unsupervised learning is in the field of density estimation in statistics, such as finding the probability density function. Though unsupervised learning encompasses other domains involving summarizing and explaining data features.

**CLUSTERING**

It is basically a type of unsupervised learning method. An unsupervised learning method is a method in which we draw references from datasets consisting of input data without labelled responses. Generally, it is used as a process to find meaningful structure, explanatory underlying processes, generative features, and groupings inherent in a set of examples. Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups. It is basically, a collection of objects on the basis of similarity and dissimilarity between them. For example, the data points in the graph below clustered together can be classified into one single group. We can distinguish the clusters,

These data points are clustered by using the basic concept that the data point lies within the given constraint from the cluster center. Various distance methods and techniques are used for calculation of the outliers.

Clustering is very much important as it determines the intrinsic grouping among the unlabeled data present. There are no criteria for a good clustering. It depends on the user, what is the criteria they may use which satisfy their need. For instance, we could be interested in finding representatives for homogeneous groups (data reduction), in finding "natural clusters" and describe their unknown properties ("natural" data types), in finding useful and suitable groupings ("useful" data classes) or in finding unusual data objects (outlier detection). This algorithm must

make some assumptions which constitute the similarity of points and each assumption make different and equally valid clusters.

### 3.5.1.1 Clustering Methods :

1. **Density-Based Methods :** These methods consider the clusters as the dense region having some similarity and different from the lower dense region of the space. These methods have good accuracy and ability to merge two clusters. Example DBSCAN

(Density-Based Spatial Clustering of Applications with Noise) , OPTICS (Ordering Points to Identify Clustering Structure) etc.

2. **Hierarchical Based Methods :** The clusters formed in this method forms a tree type structure based on the hierarchy. New clusters are formed using the previously formed one. It is divided into two categories

- Agglomerative (bottom-up approach)
- Divisive (top-down approach) .

3. **Partitioning Methods :** These methods partition the objects into k clusters and each partition forms one cluster. This method is used to optimize an objective criterion similarity function such as when the distance is a major parameter example K-means, CLARANS (Clustering Large Applications based upon randomized Search) etc.

4. **Grid-based Methods :** In this method the data space is formulated into a finite number of cells that form a grid-like structure. All the clustering operation done on these grids are fast and independent of

the number of data objects example STING (Statistical Information Grid), wave cluster, CLIQUE (Clustering In Quest) etc.

Clustering Algorithms:

- K-Means Clustering.
- Mean-Shift Clustering for a single sliding window.
- The entire process of Mean-Shift Clustering.
- DBSCAN Smiley Face Clustering.
- EM Clustering using GMMs.
- Agglomerative Hierarchical Clustering.

❖ **Semi-supervised learning**

Semi-supervised learning falls between unsupervised learning (without any labeled training data) and supervised learning (with completely labeled training data). Some of the training examples are missing training labels, yet many machine-learning researchers have found that unlabeled data, when used in conjunction with a small amount of labeled data, can produce a considerable improvement in learning accuracy.

- **Reinforcement learning**

  Reinforcement learning is an area of machine learning concerned with how software agents ought to take actions in an environment so as to maximize some notion of cumulative reward. Due to its generality, the field is studied in many other disciplines, such as game theory, control

  theory, operations research, information theory, simulation-based optimization, multi-agent systems, swarm intelligence, statistics and genetic algorithms. In machine learning, the environment is typically represented as a Markov decision process (MDP). Many reinforcement learning algorithms use dynamic programming techniques. Reinforcement learning algorithms do not assume knowledge of an exact mathematical model of the MDP, and are used when exact models are infeasible. Reinforcement learning algorithms are used in autonomous vehicles or in learning to play a game against a human opponent.

### 3.5.2 ANACONDA

Anaconda is a free and open source distribution of the Python and R programming languages for data science and machine learning related applications (large-scale data processing, predictive analytics, scientific computing), that aims to simplify package management and deployment. Package versions are managed by the package management system conda. Anaconda

Distribution is used by over 6 million users, and it includes more than 250 popular data science packages suitable for Windows, Linux, and MacOS.

Python is a high-level programming language devised by Guido van Rossum & first released in 1991. It’s the most popular coding language used by software developers to build, control, manage and for testing. It is also an interpreter which executes Python programs. The python interpreter is called python.exe on Windows.

**Python Packages**

Packages or additional libraries help in scientific computing and computational modelling. In Python, the packages are not the part of the Python standard library. Few major packages are –

NumPy (NUMeric Python): matrices and linear algebra

SciPy (SCIentific Python): many numerical routines

Matplotlib: (PLOTting LIBrary) creating plots of data

SymPy (SYMbolic Python): symbolic computation

PyTest (Python TESTing): a code testing framework

Together with a list of Python packages, tools like editors, Python distributions include the Python interpreter. Anaconda is one of several Python distributions. Anaconda is a new distribution of the Python and R data science package. It was formerly known as Continuum Analytics. Anaconda has more than 100 new packages.

This work environment, Anaconda is used for scientific computing, data science, statistical analysis, and machine learning. The latest version of Anaconda 5.0.1 is released in October 2017.The released version 5.0.1 addresses some minor bugs and adds useful features, such as updated R language support. All of these features weren’t available in the original 5.0.0 release.

This package manager is also an environment manager, a Python distribution, and a collection of open-source packages and contains more than 1000 R and Python Data Science Packages.

## IPYTHON NOTEBOOKS

IPython is a command shell for interactive computing in multiple programming languages, originally developed for the Python programming language, that offers introspection, rich media, shell syntax, tab completion, and history. IPython provides the following features:

- Interactive shells (terminal and Qt-based).
- A browser-based notebook interface with support for code, text, mathematical expressions, inline plots and other media.
- Support for interactive data visualization and use of GUI toolkits.

- Flexible, embeddable interpreters to load into one's own projects.

- Tools for parallel computing.

IPython is based on an architecture that provides parallel and distributed computing. IPython enables parallel applications to be developed, executed, debugged and monitored interactively. Hence, the I (Interactive) in

IPython.[3]This architecture abstracts out parallelism, which enables IPython to support many different styles of parallelism[4]including:
With the release of IPython 4.0, the parallel computing capabilities have been made optional and released under the ipyparallel python package. IPython frequently draw from SciPy stack[5] libraries like NumPy and SciPy, often installed alongside from one of many Scientific Python distributions. IPython provide integration some library of the SciPy stack like matplotlib, like inline graph when in used with the Jupyter notebook. Python libraries can implement IPython specific hooks to customize object Rich object display. SymPy for example implement rendering of Mathematical Expression as rendered LaTeX when used within IPython context.

**Other features**:

IPython also allows non-blocking interaction with Tkinter, PyGTK, PyQt/PySide and wxPython (the standard Python shell only allows interaction with Tkinter). IPython can interactively manage parallel computing clusters using asynchronous status call-backs and/or MPI. IPython can also be used as a system shell replacement. Its default behaviour is largely similar to Unix shells, but it allows customization and the flexibility of executing code in a live Python environment. Using IPython as a shell replacement is less common and it is now recommended to use Xonsh which provide most of the IPython feature with better shell integrations

# CHAPTER 4

# ARCHITECTURE

## 4.1 SYSTEM ARCHITECTURE

System architecture is the conceptual model that defines the structure, behaviour, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviours of the system.

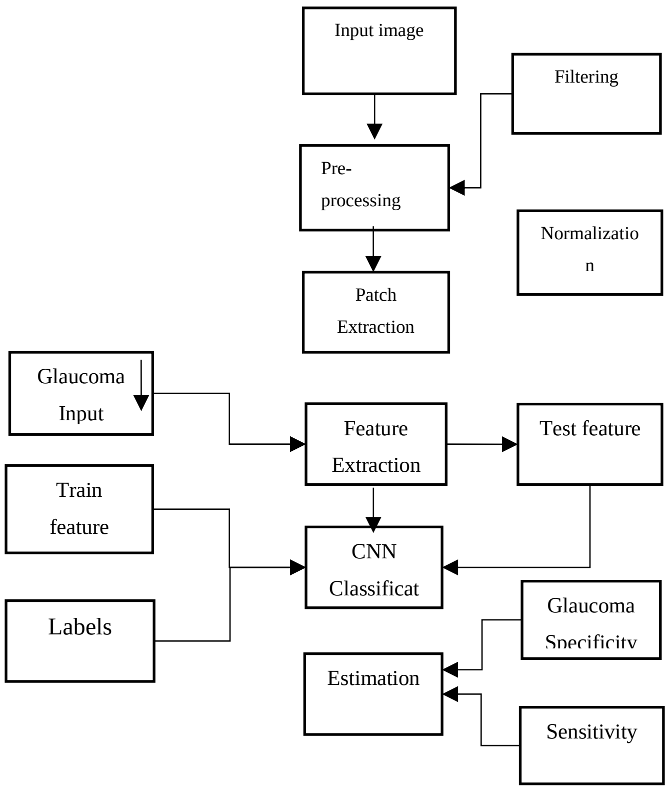


Fig 1 : System Architecture

The pre-processed image is given as an input to the CNN model which consists of an input layer, convolution layers and a fully connected layer. The input image of 256x256pixel acts as the input layer. In the first convolution layer, 16 filters of 3x3 size kernels each are applied to the input image by gliding one by one through the position and a total of 16 feature maps are generated. This method is called as feature extraction. These features are then applied to the ReLU activation function, which performs a threshold operation for each input variable with values below zero.

On the output of the ReLU layer, a max pooling layer of 2x2 window size is applied which results in down-sampling of the feature maps to 128x128 pixel. System architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

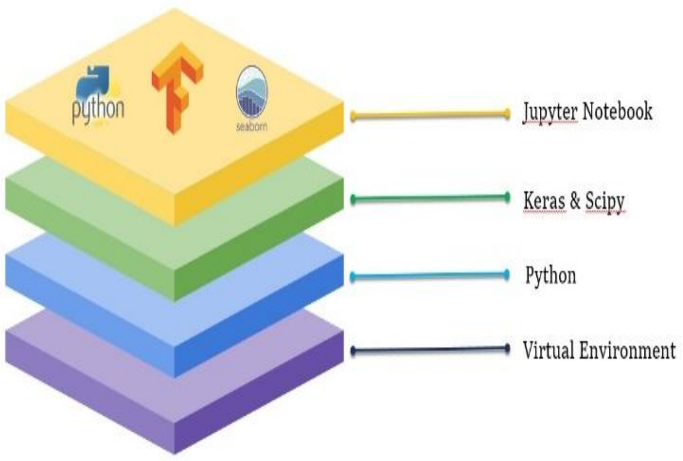


Fig 2 :Technology Stack

## 4.2 Use case  Diagram

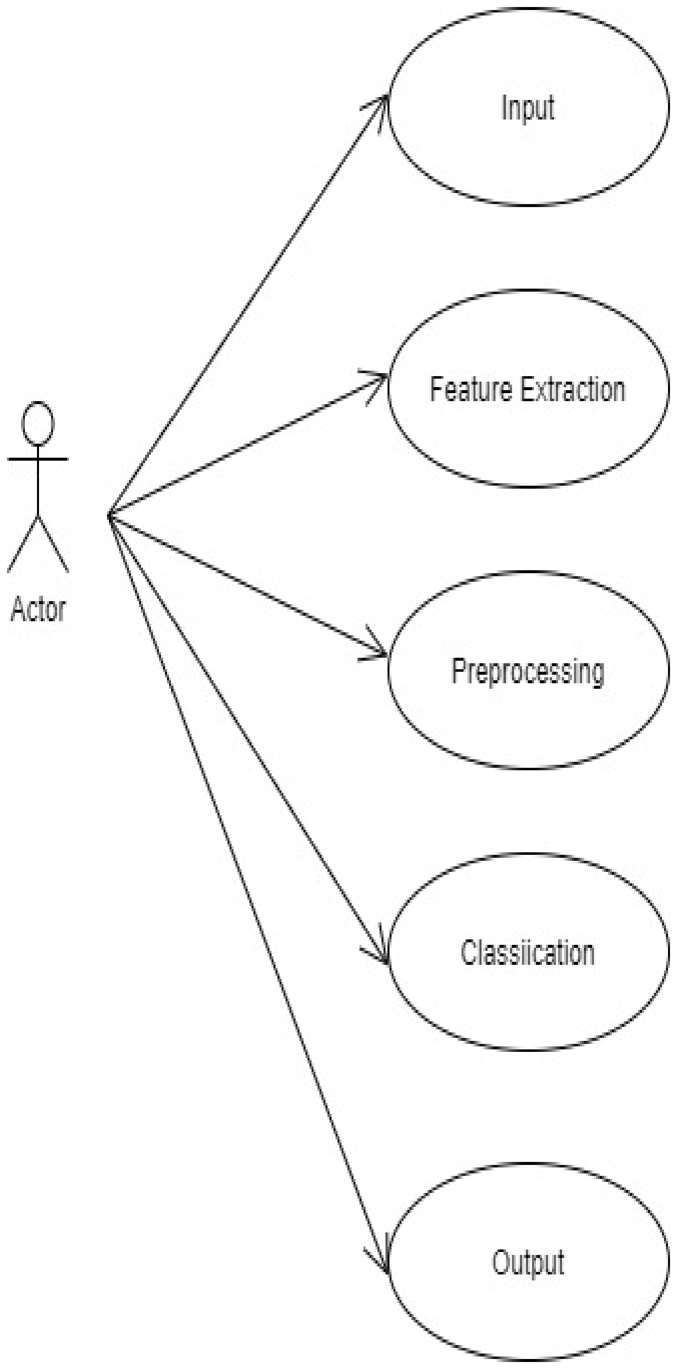


Fig 3 : Use Case Diagram

## 4.3 Activity Diagram

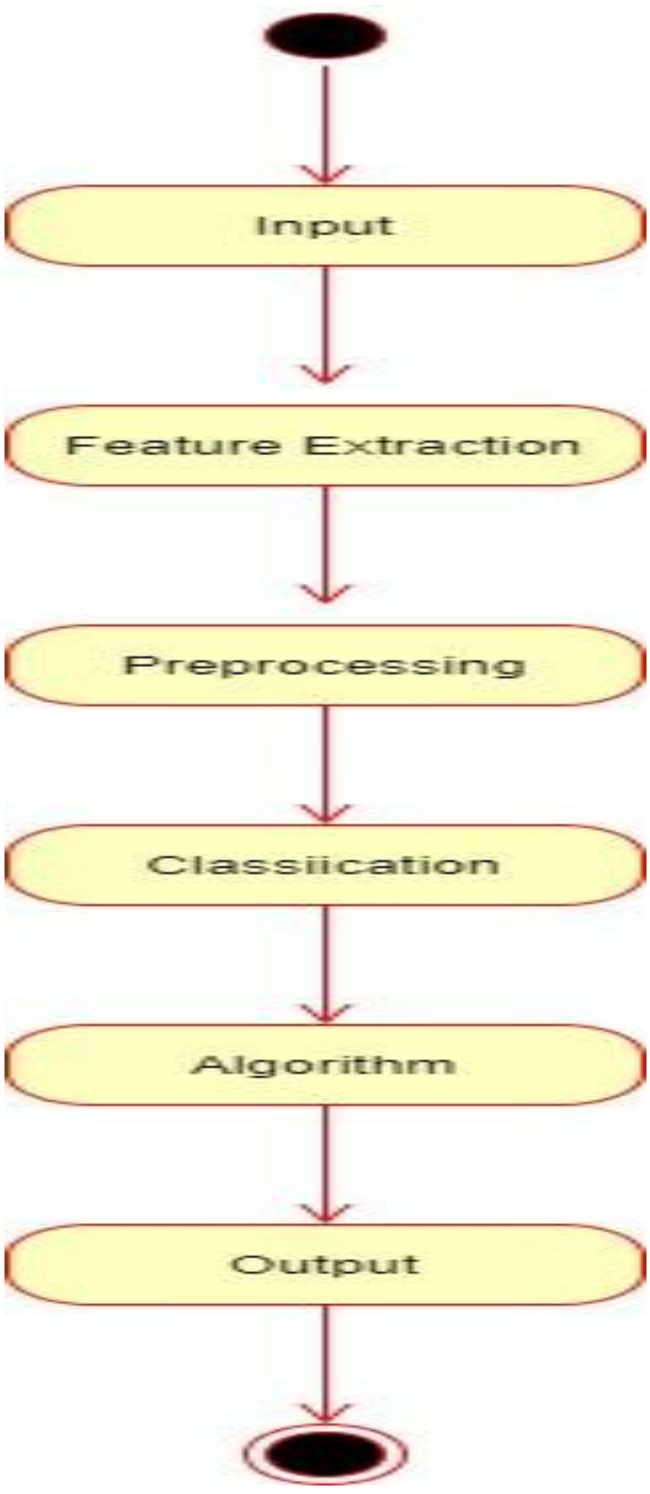


Fig 4 : Activity Diagram

## 4.4 Deployment Diagram

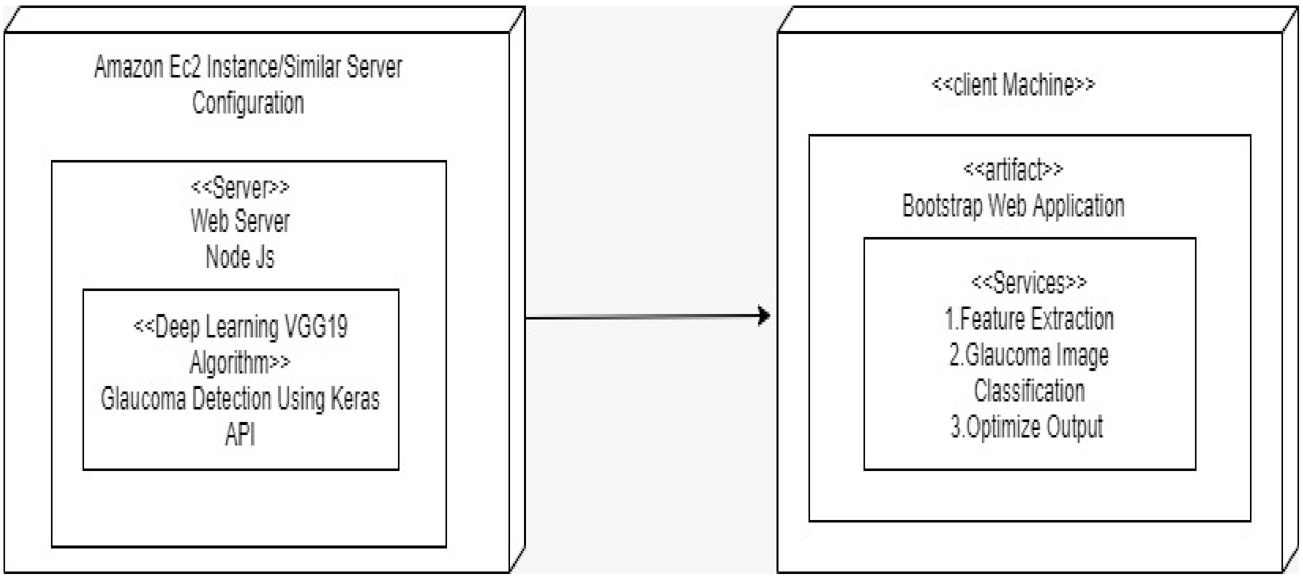

Amazon Ec2 Instance/Similar Server Configuration
<<Server>>
Web Server
Node Js
<<Deep Learning VGG19 Algorithm>>
Glaucoma Detection Using Keras API
<<client Machine>>
<<artifact>>
Bootstrap Web Application
<<Services>>
1.Feature Extraction
2.Glaucoma Image Classification
3.Optimize Output

Fig 5 : Deployment Diagram

**CHAPTER 5**

# SYSTEM MODULE

## 5.1 MODULE

- **Database**
- **Classification**
- **Datasets**

## 5.2 MODULE DESCRIPTION

### 5.2.1 Database

- Database The database contains total 2000 eye images for training and testing. Out of 2000 images, 1500 images contain glaucoma and 500 images are healthy (without glaucoma).
- All the images are of varying sizes and in different formats.

### 5.2.2 Classification

- Classification The process of classification is to classify an image according to its visual content. The classifier used is k-nearest neighbour.
- The number of neighbour (k), distance matrix and dataset is given as an input for the algorithm.

### 5.2.3 Datasets

We present a reference dataset DRISHTIGS1 consisting of 50 training and 51 testing images. For each image, manual segmentations are collected for both OD and cup region from four different human experts with varying clinical experience. Markings were collected with a dedicated marking tool which provides a fully deformable circle to enable capturing different OD and cup shapes, along with any localized shape changes such as notching. DRISHTI-GS1 is an extension of DRISHTI-GS, a dataset recently made publicly available by us [13] for OD and cup segmentation algorithms. In addition to the structural markings provided in [13], two other expert opinions are included. These are decisions on i) the image as representing a Normal or Glaucomatous eye and ii) presence or absence of notching in the inferior and/or superior sectors of the image. Since the dataset is aimed at benchmarking different segmentation algorithms, results of OD and cup segmentation methods [6,9,11] tested on this dataset are also provided. A new supervised method for notch detection based on the OD and cup segmentations is proposed. The method was evaluated on the dataset and the results are also presented. MATERIALS AND METHODS In this section, we summarize the existing datasets for ONH segmentation and present a detailed description of the presented dataset. A. Existing Datasets There are relatively few public datasets for glaucoma assessment as against a large number of datasets for diabetic retinopathy [14-16] and vessel segmentation [17,18]. Public datasets like Rim-one [8] and Drions-db [7] cater only to OD segmentation. Rim-one has 169 images (majority of them normal) with 5 manual boundary markings for each one. Drions-db has 110 images with 2 manual markings for each image. OD and cup segmentation are both important parts of ONH segmentation and together they form a base for glaucoma assessment. A reference dataset to evaluate cup segmentation methods was announced in [19]. However, it is not available for free, public access. B. Image Acquisition The dataset DRISHTI-GS1 consists of a total of 101 images. It is divided into 50 training and 51 testing images. Ground truth is provided for the training set whose details are provided in Subsection C. All images were collected at Aravind eye hospital, Madurai from visitors to the hospital, with their consent. Glaucoma patient selection was done by clinical investigators based on clinical findings during examination. Selected patients were 40- 80 years of age, with roughly equal number of males and females. Patients undergoing routine refraction test and not found to be glaucomatous were chosen to represent the normal class. All images were taken with the eyes dilated using the following data collection protocol: centered on OD with a Fieldof-View of 30-degrees and of dimension 2896 x 1944 pixels and PNG uncompressed image format. Apart from these, no other imaging constraints were imposed on the acquisition process. For each image, ground truth was collected from four glaucoma experts with experience of 3, 5, 9, and 20 years, respectively, to capture inter-observer variance in marking. Bad quality images in terms of poor contrast, positioning of OD region, etc. were discarded. In this dataset release, fundus region (image region having retinal structures) has been extracted from the original image by eliminating the surrounding non-fundus mask region, to obtain an image of approximately 2047 x 1760 pixels.

A dedicated marking tool was developed to obtain boundary marking on images from human experts. This tool allows precise boundary marking for even irregular shapes of OD and cup regions. This was favored over approximating the boundary with some parametric shape such as ellipse, as local deformations in the cup are possible and need to be captured in the marking. A deformable circle with several free-to-move control points was provided to the user. These help the user to first position the circle close to the approximate region boundary and next do a better fit to the region by moving the individual control points. Thus, an accurate shape marking is possible in two steps: rough localization of the circle followed by local deformations to the circle. Figure 2 shows a screenshot of the tool wherein one point in the OD boundary is shown in pre-final position. Additionally, diagnostic opinion on each image being Normal or Glaucomatous was obtained from the 4 glaucoma experts and a Gold standard was derived based on the majority opinion, i.e., 3 out of the 4 experts. Further, image level decision on presence of notching in the Inferior and Superior sectors was also obtained from the most experienced glaucoma expert. The ground truth is released only for the 50 training images. In addition to the raw dataset published in this journal, a web page has been designed to allow interested research groups to register, download the test set and submit segmentation results (for OD and cup) on the same. The submitted results will be analyzed using evaluation measures presented in Subsection D and made publicly available on DRISHTI-GS web page

With the implantation of TensorFlow Framework and Keras model for deep learning and training the model, the level of precision of the Glaucoma detection has unavoidably improved. When an image is uploaded into the system, it is compared to the model that has already been trained and then processed.

In the implementation of the glaucoma the images can be classified into the two dataset such as the images with glaucoma and images without glaucoma

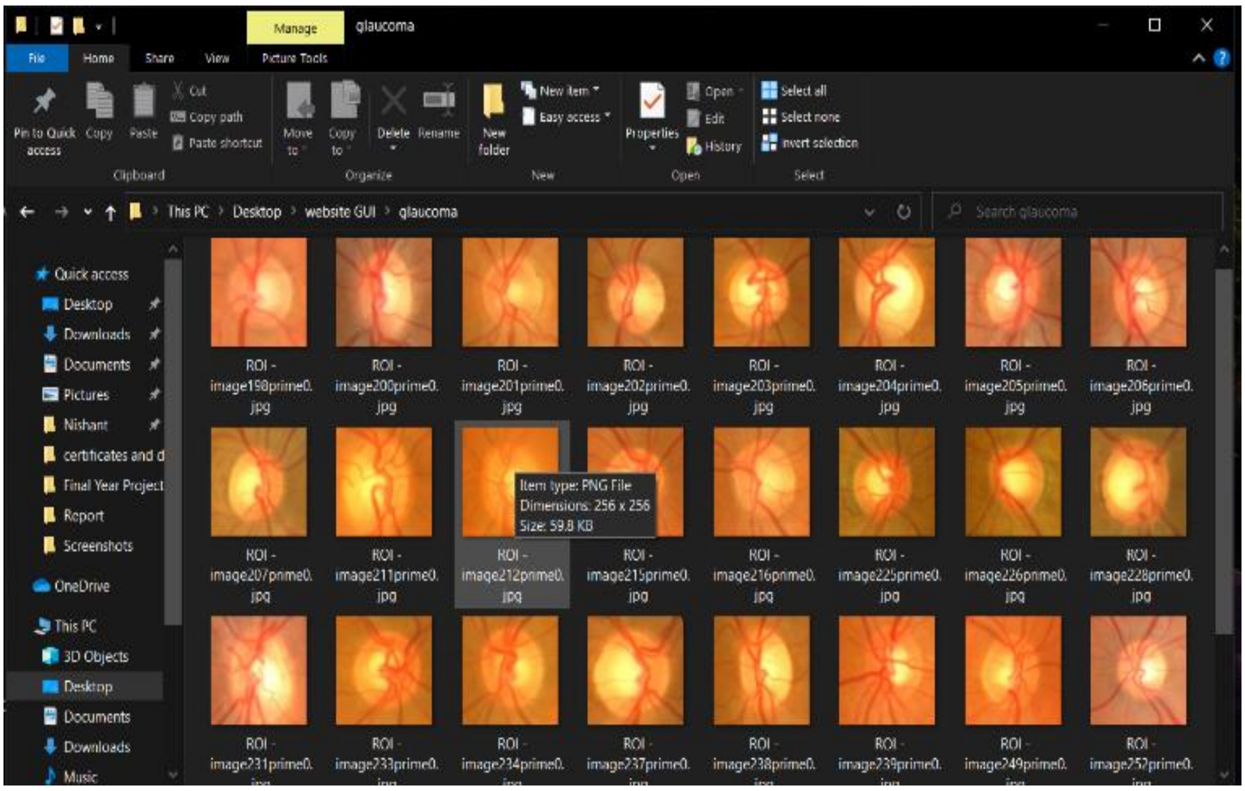


Fig 6 : Sample Datasets Without Glaucoma

While processing the input portraits, the system compares them to existing datasets of over 2000 glaucoma and non-glaucoma snaps and provides an accurate result.

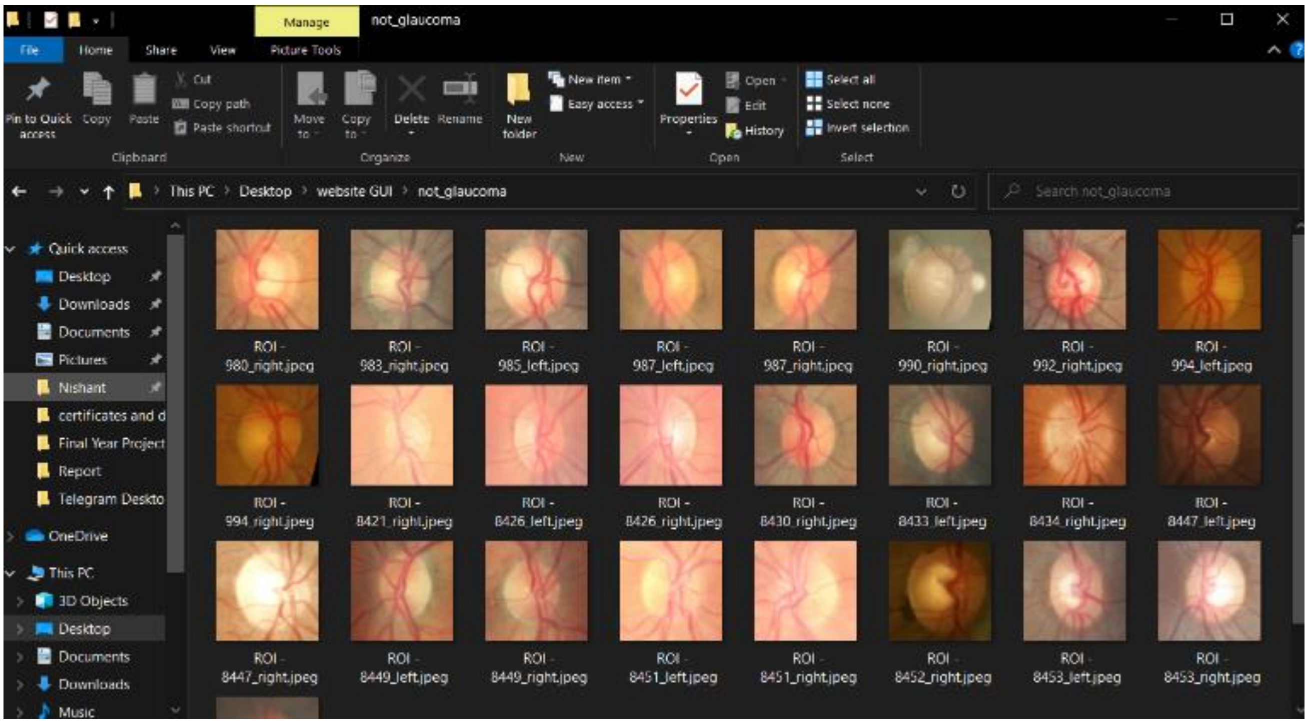


Fig 7 : Sample Datasets With Glaucoma

# CHAPTER 6

# SYSTEM IMPLEMENTATION

## 6.1 CLIENT-SIDE CODING

### Home Module Html

```
 <!DOCTYPE html>
<html lang="en">
  <head>
    <title>Galucoma Detection</title>
    <meta charset="utf-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <!-- Loading stylesheets: -->
    <link
      rel="stylesheet"
      href="https://fonts.googleapis.com/css?family=Barlow:500"
    />
    <link rel="stylesheet" href="./css/reset.css" />
    <link rel="stylesheet" href="./css/style.css" />
    <!-- loading the teachable machine libraries: -->
    <!-- more documentation at
https://github.com/googlecreativelab/teachablemachine-libraries -->
    <script
src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@1.3.1/dist/tf.min.js"></script>
    <script
src="https://cdn.jsdelivr.net/npm/@teachablemachine/image@0.8/dist/teachablemach
ine-image.min.js"></script>
```

```
  <!-- these load our code: -->
  <script type="module" src="./js/model-runner.js"></script>
  <script type="module" src="./js/bar-graph.js"></script>
</head>
<body>
  <section id="model">
    <h1>Glaucoma Detection for your EYE</h1>
    <h2>Show Your Face</h2>
    <div id="webcam-wrapper">
      <div class="loader"></div>
    </div>
    <div id="graph-wrapper"></div>
  </section>
  <section id="info">
    <h2>
      This model find Glucoma or not
    </h2>
    <h2>
      <a href="h"></a>
    </h2>
  </section>
  <footer>
   <h2>  <a> </a >
   </h2>
  </footer>
  <script type="module">
```

```
      import { setupModel } from "./js/model-runner.js";
      import { setupBarGraph, updateBarGraph } from "./js/bar-graph.js";
      let URL = "https://teachablemachine.withgoogle.com/models/p1IzQvtOT/";
      // setupBarGraph is defined in the js/bar-graph.js file
      setupBarGraph(URL);
      // setupModel is defined in the js/model-runner.js file
      // we pass in another function: updateBarGraph (defined in the js/bar-graph.js file)
      // setupModel will store the updateBarGraph function,
      // and call it every time it has new prediction data from the model
      setupModel(URL, data => {
        updateBarGraph(data);
      });
    </script>
  </body>
</html>
```

### 6.1.1 SCRIPT FILES

### Bar Graph Java Script

```
let labels = [];
let bars = {};
let graphWrapper;
// these are the colors of our bars
let colors = ['#E67701', '#D84C6F', '#794AEF', '#1291D0'];
let lightColors = ['#FFECE2', '#FFE9EC', '#F1F0FF', '#E2F5FF'];
// This function makes the bar graph
// it takes in a URL to a teachable machine model,
// so we can retrieve the labels of our classes for the bars
```

```
export async function setupBarGraph(URL) {
    // the metatadata json file contains the text labels of your model
    const metadataURL = `${URL}metadata.json`;
    // get the metadata fdrom the file URL
    const response = await fetch(metadataURL);
    const json = await response.json();
    // get the names of the labels from the metadata of the model
    labels = json.labels;
    // get the area of the webpage we want to build the bar graph
    graphWrapper = document.getElementById('graph-wrapper');
    // make a bar in the graph for each label in the metadata
    labels.forEach((label, index) => makeBar(label, index));
}
// This function makes a bar in the graph
function makeBar(label, index) {
    // make the elements of the bar
    let barWrapper = document.createElement('div');
    let barEl = document.createElement('progress');
    let percentEl = document.createElement('span');
    let labelEl = document.createElement('span');
    labelEl.innerText = label;
    // assemble the elements
    barWrapper.appendChild(labelEl);
    barWrapper.appendChild(barEl);
    barWrapper.appendChild(percentEl);
    let graphWrapper = document.getElementById('graph-wrapper');
    graphWrapper.appendChild(barWrapper);
    // style the elements
    let color = colors[index % colors.length];
    let lightColor = lightColors[index % colors.length];
    barWrapper.style.color = color;
    barWrapper.style.setProperty('--color', color);
    barWrapper.style.setProperty('--color-light', lightColor);
    // save references to each element, so we can update them later
    bars[label] = {
        bar: barEl,
        percent: percentEl
    };
}
// This function takes data (retrieved in the model.js file)
// The data is in the form of an array of objects like this:
// [{ className:class1, probability:0.75 }, { className:class2, probability:0.25 }, ... ]
// it uses this data to update the progress and labels of of each bar in the graph
export function updateBarGraph(data) {
    // iterate through each element in the data
    data.forEach(({ className, probability }) => {
        // get the HTML elements that we stored in the makeBar function
        let barElements = bars[className];
        let barElement = barElements.bar;
        let percentElement = barElements.percent;
        // set the progress on the bar
        barElement.value = probability;
```

```
            // set the percent value on the label
            percentElement.innerText = convertToPercent(probability);
        });
    }
    // This function converts a decimal number (between 0 and 1)
    // to an integer percent (between 0% and 100%)
    function convertToPercent(num) {
        num *= 100;
        num = Math.round(num);
        return `${num}%`;
    }
```

## Module Runner Java Script

```
     let model, webcam, predictionCallback;
    // This function sets up the model trained in Teachable Machine.
    // it takes in the URL to the model, and a function to be run
    // each time the model makes a new prediction.
    export async function setupModel(URL, predictionCB) {
        //store the prediction callback function
        predictionCallback = predictionCB;
        // the model.json file stores a reference to the trained model
        const modelURL = `${URL}model.json`;
    // the metatadata.json file contains the text labels of your model and additional
information
    const metadataURL = `${URL}metadata.json`;
    // Load the model using the tmImage library
    model = await window.tmImage.load(modelURL, metadataURL);
    // this function from the tmImage library returns a video element that
    // shows a video feed from the webcam
    webcam = new window.tmImage.Webcam(200, 200, true); //width, height, flipped
    await webcam.setup(); // request access to the webcam
    await webcam.play();
    // add the video element to the page
    document.getElementById('webcam-wrapper').appendChild(webcam.canvas);
    // kick off the model prediction loop
    window.requestAnimationFrame(loop); }
// This function will run forever in a loop
async function loop() {
    // update the webcam frame
    webcam.update();
    // make a prediction using the model
    await predict();
    // then call loop again
    window.requestAnimationFrame(loop);
}
// This function uses the model we loaded to make a prediction on the webcam data
async function predict() {
    // predict can take in an image, video or canvas html element
    const prediction = await model.predict(webcam.canvas);
    // Call the prediction callback function now that we have new prediction data
    predictionCallback(prediction);
}
```

## 6.2 SERVER-SIDE CODING

```
import tensorflow.keras
from PIL import Image, ImageOps
import numpy as np

# Disable scientific notation for clarity
np.set_printoptions(suppress=True)

# Load the model
model = tensorflow.keras.models.load_model('keras_model.h5')

# Create the array of the right shape to feed into the keras model
# The 'length' or number of images you can put into the array is
# determined by the first position in the shape tuple, in this case 1.
data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)

# Replace this with the path to your image
image = Image.open('test.png’)

#resizing the image to be at least 224x224 and then cropping from the center
size = (224, 224)
image = ImageOps.fit(image, size, Image.ANTIALIAS)
```

```
#turn the image into a numpy array
image_array = np.asarray(image)

# display the resized image
image.show()

# Normalize the image
normalized_image_array = (image_array.astype(np.float32) / 127.0) - 1

# Load the image into the array


data[0] = normalized_image_array

# run the inference
prediction = model.predict(data)
#print(prediction)
```

# CHAPTER 7

# SYSTEM TESTING

## 7.1 TESTING TECHNIQUES

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as-yet – undiscovered error. A successful test is one that uncovers an as-yet- undiscovered error. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences. It verifies that the whole set of programs hang together. System testing requires a test consists of several key activities and steps for run program, string, system and is important in adopting a successful new system. This is the last chance to detect and correct errors before the system is installed for user acceptance testing.

The software testing process commences once the program is created and the documentation and related data structures are designed. Software testing is essential for correcting errors. Otherwise, the program or the project is not said to be complete. Software testing is the critical element of software quality assurance and represents the ultimate the review of specification design and coding. Testing is the process of executing the program with the intent of finding the error. A good test case design is one that as a probability of finding a yet undiscovered error. A successful test is one that uncovers a yet undiscovered error. Any engineering product can be tested in one of the two ways:

### 7.1.1 WHITE BOX TESTING

This testing is also called as Glass box testing. In this testing, by knowing the specific functions that a product has been design to perform test can be conducted that demonstrate each function is fully operational at the same time searching for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test cases. Basis path testing is a white box testing.

Basis path testing:

- Flow graph notation
- Kilometric complexity
- Deriving test cases
- Graph matrices Control

### 7.1.2 BLACK BOX TESTING

In this testing by knowing the internal operation of a product, test can be conducted to ensure that “all gears mesh”, that is the internal operation performs according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software.

The steps involved in black box test case design are:

### 7.1.3 SOFTWARE TESTING STRATEGIES:

A software testing strategy provides a road map for the software developer. Testing is a set activity that can be planned in advance and conducted systematically. For this reason, a template for software testing a set of steps into which we can place specific test case design methods should be strategy should have the following characteristics:

Testing begins at the module level and works “outward” toward the integration of the entire computer-based System.

## 7.2 Testcases and Reports

| TestCaseId | Action | Expected Result | Actual Result | Status |
|---|---|---|---|---|
| 1. | Accessing the webcam | The UI request the OS to access webcam | The UI request the OS to access webcam | pass |
| 2. | PERSON WITH GLAUCOMA problem | The UI glaucoma progress bar should increase | The UI glaucoma progress bar increases | pass |
| 3. | Person without glaucoma problem | The UI without glaucoma progress bar should increase | The UI without glaucoma progress bar increases | pass |

Fig 1 : Testcase Table

# CHAPTER 8

# SCREENSHOT

## 8.1 DATASETS SAMPLE IMAGES

### With Glaucoma

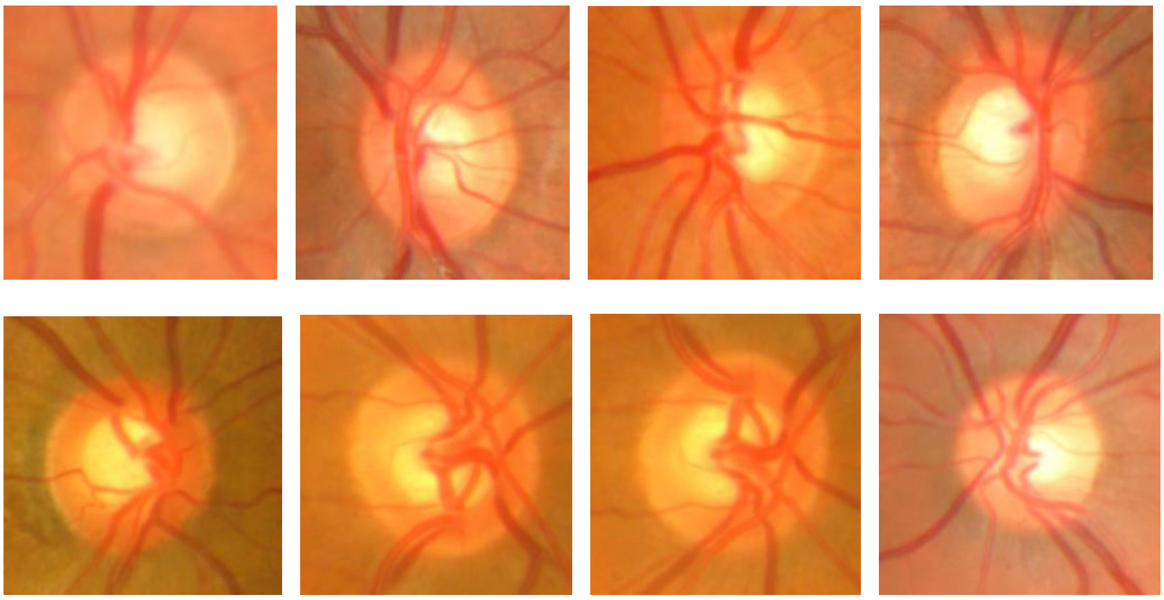


Fig 8 : With Glaucoma

### Without Glaucoma

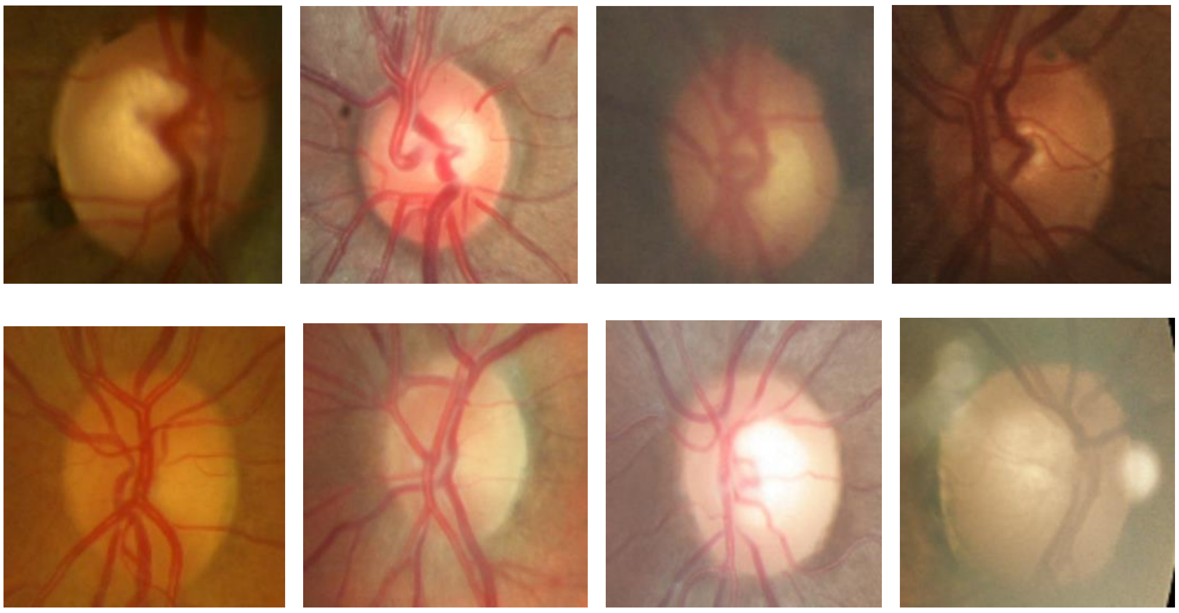
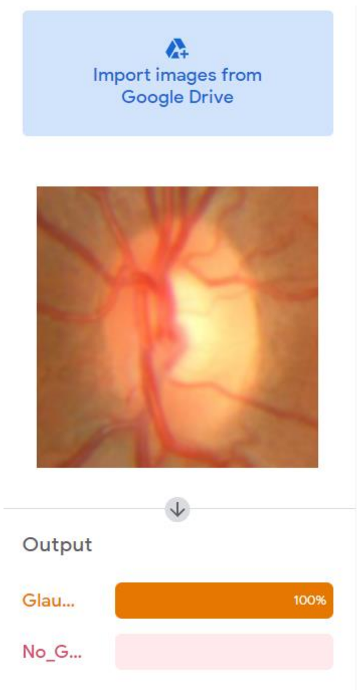


Fig 9 :Without Glaucoma

## 8.2 OUTPUTS

**With Glaucoma** **Without Glaucoma**



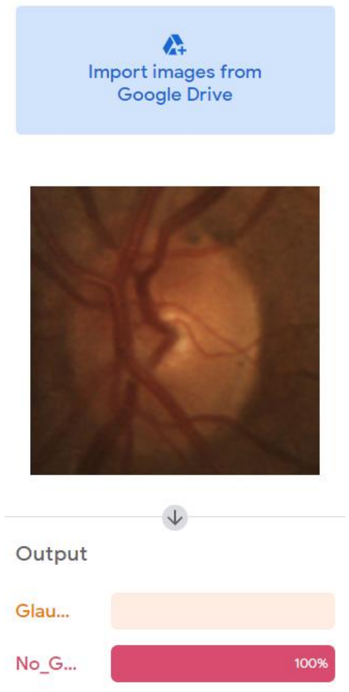


Fig 10 : Output Screenshot_1

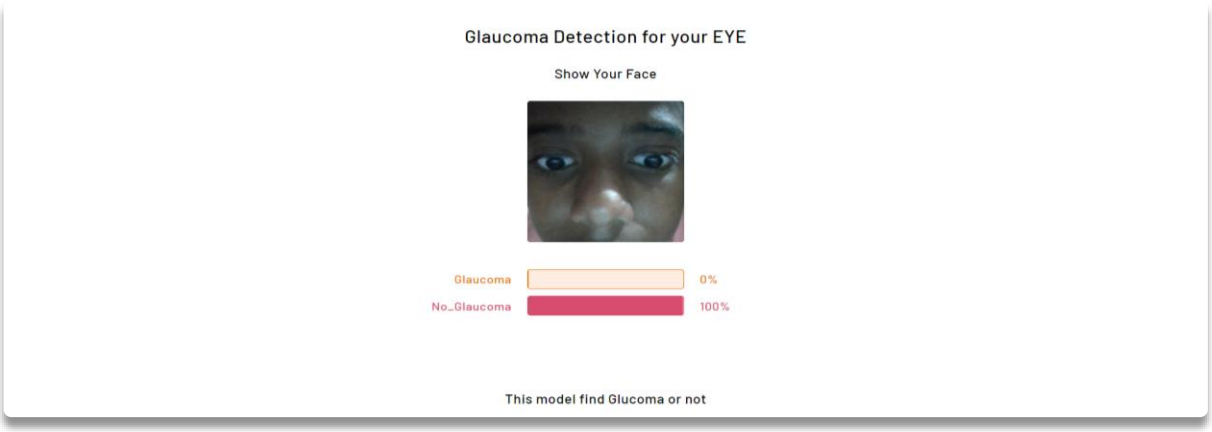

Glaucoma Detection for your EYE
Show Your Face
Glaucoma
0%
No_Glaucoma
100%
This model find Glucoma or not

Fig 11 : Output Screenshot_2

# CHAPTER 9

# CONCLUSION

Glaucoma is a “silent thief of sight” having no early symptoms and can cause permanent blindness if not detected or diagnosed at an early stage. Fundoscopy enables ophthalmologists to analyze the internal retinal structural changes i.e. change in CDR, ISNT ratio. Regular retinal layer analysis is a fundamental need to prevent glaucoma or stop glaucoma progression. Research is being done in biomedical imaging to propose algorithms for CAD systems which aids doctors in analyzing and screening the affected patients. Many autonomous glaucoma detection systems help in early detection of glaucoma by analyzing the structural changes in the internal retina. CDR, ISNT ratio, NRR ratio, Vertical and horizontal Cup height are the fundamental structural changes that appear in case of glaucoma progression and are being analyzed by all the autonomous glaucoma detection systems. Many states of art Machine learning techniques are also being used in glaucoma detection systems which uses textural and intensity-based features to discriminate healthy eyes from glaucomatous eyes. Proposed methodology provides an algorithm to detect glaucoma by analyzing the structural changes from fundus image and correlate the results with classification results from machine learning module. This hybrid of structural changes-based evaluation and machine learning based evaluation results in more accurate results and improves the sensitivity to 1. Images labeled as healthy or non-healthy by both the modules are classified as healthy or non-healthy respectively. If the results from both modules do not converge at a decision, image is classified as suspect. Glaucoma detected cases and suspects are referred to ophthalmologists for further examination and medication. Proposed system is able to screen out glaucoma patients 100 % accuracy as none of the glaucoma case is classified as normal.

# REFERENCE

[1]“Detection of Glaucoma Using Retinal Fundus Images” Hafsah Ahmad, Abubakar Yamin, Aqsa Shakeel, Syed Omer Gillani, Umar Ansari(IEEE 2014) .

[2]“Automatic Glaucoma Detection by Using Funduscopic Images”, Atheesan S., Yashothara S.(IEEE 2016).

[3]”Automated Detection of Suspected Glaucoma in Digital Fundus Images”, Namita Sengar, Malay Kishore Dutta, Radim Burget, Martin Ranjoha(IEEE 2017) .

[4] Budai A, Bock R, Maier A, Hornegger J, Michelson G. Robust vessel segmentation in fundus images. Int J Biomed Imag. 2013.

[5]Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. 2014. ArXiv e-prints arxiv:abs/1409.1556.

[6]Carneiro G, NascimentoJ, Bradley AP. In : Navab N, Hornegger J, Wells WM, Frangi AF, eds. Unregistered multiview mammogram analysis with pre-trained deep learning models.

[7]World Health Organization. Bulletin of the World Health Organization, Volume 82(11). 2004.

[8] Bourne RRA. Worldwide glaucoma through the looking glass. Br J Ophthalmol.2006.

[9]Bock R, Meier J, Nyúl LG, Hornegger J, Michelson G. Glaucoma risk index: automated glaucoma detection from color fundus images. Med Image Anal. 2010.