DIAGNOSTIC OF EYE ILLNESSES (GLAUCOMA & ARMD)

[1]M Ramya, Assistant Professor, Mail ID:bvenkat1109@gmail.com
[2]Dr M Bal Raju, Professor, Mail ID:drrajucse@gmail.com
College Name: Swamy Vivekananda Institute of Technology
[4]Dr.Palla.Karunakar Reddy, Associate Professor, Mail ID:@gmail.com
Department of CSE Engineering,
Pallavi Engineering College Hyderabad, Telangana 501505.

ABSTRACT

As population aging has become a major demographic trend around the world ,patients suffering from eye diseases , such as Glaucoma, ARMD are expected to increase. Early detection and appropriate treatment of eye diseases are of great significance to prevent vision loss and promote living quality. Conventional diagnosis methods are tremendously dependent on physicians, professional experience and knowledge, which lead to high misdiagnosis rate and huge waste of medical data.

In this project, a deep learning model based method which is inspired by the diagnostic process of human ophthalmologists is proposed to automatically classify the fundus photographs into 2 types with or without ARMD categories also, with Or without Glaucoma. The project consists of two different neural network models developed to recognize the diseases, Glaucoma and ARMD.Better accuracy is obtained as we use deep learning. This project will be an aid to eye specialists in giving an efficient treatment. Eyesight is one of the most important senses, the developed project can help people all over to maintain eye care.

This project uses Kaggle Glaucoma and ARMD datasets.This model predicts Glaucoma with 90% accuracy and ARMD with more than 70% accuracy.

1. INTRODUCTION

The rising prevalence of age-related eye diseases, particularly age-related macular degeneration, places an ever-increasing burden on health care providers. As new treatments emerge, it is necessary to develop methods for reliably assessing patients' disease status and stratifying risk of progression. The presence of drusen in the retina represents a key early feature in which size, number, and morphology are thought to correlate significantly with the risk of progression to sight-threatening age-related macular degeneration. The damage of optic nerve which is especially responsible for the proper eyesight, if damaged causes Glaucoma. Glaucoma is not only caused for elderly people, but also can affect people of any age. Manual labeling of drusen or any damage in the eye, on color fundus photographs by a human is labor intensive and is where automatic computerized detection would appreciably aid patient care.

This project aims to develop appropriate algorithm to detect these diseases. The retinal or fundus images of eye, are processed to recognize any symptoms of diseases, such as Glaucoma, ARMD(Age-related macular degeneration).

Based on the symptoms, the disease is identified, if any. This project will be an aid to eye specialists in giving an efficient treatment. Eyesight is one of the most important senses, the developed project can help people all over to maintain eye care.

1.1 Problem Definition:

Eye sight is one of your most important senses: 80% of what we perceive comes through our sense of sight. Early detection and appropriate treatment of eye diseases are of great significance to prevent vision loss and promote living quality.The aim of the project is to develop a deep learning model which uses Convolutional Neural Networks to detect whether the given image of fundus is suffering from Glaucoma or Not and ARMD or not.Detection of Eye Diseases i.e., Glaucoma & ARMD model is built using Keras API of Tensorflow 2.0. The deep learning techniques will aid in fast and accurate diagnosis.

1.2 Project Scope:

This project aims to detect the ARMD and GLAUCOMA diseases using deep learning model. In this project, a novel convolutional neural network model with the CNN architecture is trained with a transfer learning technique. The proposed model accepts fundus images as inputs and learns from their features to help to make a prediction. This project is developed with a motive to help Eye care workers, to help reduce the time taken for generating a diagnosing report of disease recognition, which would manually take upto 2 weeks of time. This system can be used for mass Eye care camps across remote areas, which would hugely benefit the people living in such areas.

1.3 Existing System:

Most of diagnosis processes suggested earlier either rely on the variables manually measured by experts or put much effort into extracting handcrafted features with image processing approaches which bring extra complexity and instability. Thus, the deep-learning method with the ability to learn significant features directly from the fundus photography has aroused the attention of researchers in recent years.

Disadvantages:

- Uses machine learning approach which depends on clinical relevant variables such as micro count and abnormality in the graded retinal images labeled by human experts.
- Time Consuming

Proposed System:

In this project, a deep learning model based method which is inspired by the diagnostic process of human ophthalmologists is proposed to automatically classify the fundus photographs into 2 types with or without ARMD categories also, with Or without Glaucoma.Quellecet al. proposed a system to detect referable ARMD by employing a deep .convolutional neural network (CNN) and automated segment ARMD lesions by creating heat maps of the convolutional layer which shows the potential to discover new bio markers in images.

The project consists of two different neural network models developed to recognize the diseases, Glaucoma and ARMD. The trained deep learning model extracts the features from the images dataset and learns to recognize the disease by making the prediction. When a fundus eye image is given as input, it categorizes into two categories for each disease (i.e., Glaucoma, Non Glaucoma and ARMD, Non ARMD).Better accuracy is obtained as we use deep learning.

The proposed system consists of the following goals and advantages:

Goals:

- To build Effective model to predict Glaucoma and ARMD.
- Classification and prediction of Glaucoma and ARMD.
- To get high accuracy and precision.

Advantages:

- It doesn’t require high computational power.
- It is very easy to implement.
- It is easily interpretable.
- It is very efficient
- It outputs well-calibrated predicted probabilities.
- High accuracy and precision.
- The database system is fast and can handle large data sets.

1.5 Requirements Specification :

1.5.1 Software Requirements :

The following Table contains the software requirements for Detection of Glaucoma and ARMD using CNN :

Software Requirements

OS	:	Windows
Python IDE	:	python 3.x and above
Coding platform:		Google Colab
Datasets	:	Kaggle, Stare
GUI	:	ANVIL Application

Table 1.1 : Software Requirements

1.5.2 : Hardware Requirements :

The following Table contains the hardware requirements for Detection of Glaucoma and ARMD using CNN :

Hardware Requirements

RAM	:	4GB and Higher
Processor	:	Intel i3 and above
Hard Disk	:	500GB: Minimum

Table 1.2 : Hardware Requirements

1.6 Software Tools Used:

1.6.1 Python:

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Python is simple and easy to learn. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed. Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy and a bug or bad input will never cause a segmentation fault[5]. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on[5]. The debugger is written in Python itself, testifying to Python's introspective power. The Proposed System works on python 3.5 and above.

- **Python is Interpreted − Python is processed at runtime by the interpreter. You do not need** to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive − You can actually sit at a Python prompt and interact with the** interpreter directly to write your programs.
- Python is Object-**Oriented − Python supports Object**-Oriented style or technique of programming that encapsulates code within objects.

This project uses Python language because it consists of various powerful and efficient Deep Learning modules which can be used to get deep insights from the data and perform accurate predictions.

1.6.2 CNN(Convolutional Neural Networks):

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms.

The Deep Learning Models built in this project uses Conv2D layer i.e., convolutional layer as a part of its neural network. This project uses two layers of convolution as of part of its model to extract the important features from the data by removing noise and unwanted features from images.

1.6.3 Pandas :

Pandas is the most popular python library that is used for data analysis. It provides highly optimized performance with back-end source code is purely written in C or Python. This package is used along with matplotlib to perform operations on images in data pre-processing.

1.6.4 NumPy :

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.It is the fundamental package for scientific computing with Python.

This project uses NumPy for image pre-processing. The images are converted to pixels or arrays using NumPy module. Also, the user input i.e., fundus image is converted to array using this module and also used to perform numerical operations on image arrays like resizing etc.,

1.6.5 Matplotlib :

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John

Hunter in the year 2002.One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

This project uses Matplotlib library to plot a graph on accuracy and loss of our Deep Learning Models.

1.6.6 Keras API :

Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result as fast as possible is key to doing good research.

In this project, the Keras API is used to create layers of Deep Learning Model like Conv2D, MaxPooling, Flatten, Dense and to build Sequential Model. Keras API is also used to perform operations on image use its image module to load image, resize image etc.,

1.6.6 Tensorflow2 :

Currently, the most famous deep learning library in the world is Google's TensorFlow. Google product uses machine learning in all of its products to improve the search engine, translation, image captioning or recommendations. To give a concrete example, Google userscan experience a faster and more refined the search with AI. If the user types a keyword a the search bar, Google provides a recommendation about what could be the next word.Google wants to use machine learning to take advantage of their massive datasets to give users the best experience.

TensorFlow is a library developed by the Google Brain Team to accelerate machine learning and deep neural network research. It was built to run on multiple CPUs or GPUs and even mobile operating systems, and it has several wrappers in several languages like Python, C++ or Java. Tensorflow architecture works in three parts:

- Preprocessing the data
- Build the model
- Train and estimate the model

1.6.7 Google Colab:

Colab is a free Jupyter notebook environment that runs entirely in the cloud. Most importantly, it does not require a setup and the notebooks that one create can be simultaneously edited by other team members - just the way how we can edit documents in Google Docs. Colab supports many popular machine learning libraries which can be easily loaded in the notebook.

This Colab notebook is similar to Jupyter notebook with all the powerful libraries of Machine Learning , Deep Learning are readily available. Also this is a GUI interface and web based IDE, there is no requirement to install it in our local machine and it is free to use. In this project, Colab notebook works as our server i.e., backend part which accepts user input from anvil front-end application and processes it and predicts the output.

1.6.8 : ANVIL :

Anvil is a platform for building and hosting full-stack web **apps** written entirely inPython. Drag & drop your UI, then write Python on the front-end and back-end to make it all work. Web development has never been this easy (or fast)! **Anvil** is a tool in the Platform as a Service category of a tech stack.

This project uses Anvil platform to convert our colab notebook as web application i.e., the user interface for this project is created on Anvil platform. The project uses Anvil packages like anvil.server, anvil.media and anvil.server.connect to interact with our colab notebook i.e., server side.

The various features of the Anvil platform are :

- Build full – stack web apps with nothing but Python.
- It is a web – based IDE no need to install it.
- We can create our User Interface with drag and drop.
- We can run Python code on server.
- It has built-in database.
- Built-in user authentication.
- Can generate PDFs.

2. LITERATURE SURVEY

Horta et al. reported a hybrid method- frame employing deep image features and random forest to combine di erent patient non-visual data like lifestyle, cataract, demographics with the image for classi cation. Detec-tion of the circular boundaries of the retina, normalization of intensity levellightness channel extraction from L*a*b color space and nally resizing the image for changing the resolution was done as preprocessing steps. Next, to extract deep image features, CNN (pre-trained with 1.2 million image data) was used. The deep features combined with the non-medical non-visual in-formation of the patients were used to train a Random Forest Classier to perform binary classification for higher severity AMD and lower severity of AMD.

Govindaiah et al.[71] reported an extended study of previous work on AREDS fundus AMD dataset using a modi ed VGG16 architecture. Macula was chosen as a Region of Interest and images were resized to a common reference level. For classi cation, a modi ed VGG16 architecture was implemented using 3x3 convolution layer and 2x2 pooling layer and for comparison a 50 layer Keras implementation of residual neural network was used. This architecture gave an accuracy of upto 92.5% in binary classification between no/early stage AMD and intermediate/advanced stage ARMD.

Chen et al.[79]implemented a CNN with dropout and data augmentation on ORIGA and SCES dataset. A six layers deep CNN with 4 convolutional layers of progressively decreasing lter size (11, 5, 3, 3) followed by 2 dense layers was used to get 83.1% and 88.7% AUC on ORIGA and SCES respectively.

Table 2.1 shows the literature survey of Traffic Accidents Classification and Prediction using Logistic Regression. It contains name of author, title of proposed work, method, accuracy, objective, results of various existing systems.

S.No	Author	Title	Dataset	Method	Accuracy	Demerits
1	Horta et al.	ARMD Detection using DCNN	AREDS	DCNN	76.9%	Used patients non-visual Data which may affect the prediction .
2	Govindaiah	ARMD Detection using	AREDS	VGG16	90%	Very much time taking To train the system.
3	Fu et al.	Glaucoma Identification Using DENet	DENet	SCES	83.2%	Complex process.
4	Fu et al.	Glaucoma Detection	DENet	SINDI	66.6%	Complex and Less accurate.
5	Muhammad et al.	Glaucoma Detection	HDLM	Private 102 images	93.1%	Use of private dataset and less number of images. The accuracy obtained is not accepted.

Table 2.1: Literature survey of Detection of Glaucoma and ARMD.

2.1Theoretical Analysis of ARMD :

Automated screening tool for ARMD :

Aging is the prime cause of age-related macular degeneration (ARMD). There are primarily two types of ARMD: (i) dry, and (ii) wet. The dry ARMD is the common form of ARMD. It typically starts with the formation of small pale yellowish deposits called drusen under the retina, causing atrophy at the macula, which is known as age-related macular degeneration. This, in turn, affects the central vision of a person. The wet ARMD is caused due to the abnormal growth of blood vessels under the retina. These vessels are known as the choroid neo vascular membrane break through the retina and bleed. They finally lead to either scarring at the macula or atrophic changes leading to severe visual impairment. Hence, the wet ARMD progresses faster and leads to an irreversible loss of sight. Therefore, it is advisable to go for routine eye screening, especially for elderly subjects. Although the ARMD cannot be fully cured, an accurate early detection can impede the progression of vision loss. However, manual diagnosis of ARMD is difficult and subjective. Thus, a computer-aided diagnosis (CAD) system can be utilized to automatically screen the eyes and give an accurate diagnosis of the type of ARMD. In this study, we propose a novel technique to identify normal, dry, and wet ARMD. A total of 945 fundus images (404 normal, 517 dry ARMD, and 24 wet ARMD) are used in this proposed framework.

The Pyramid of Histograms of Orientation Gradients (PHOG) technique is implemented in this work to capture the subtle changes in the pixels of fundus images. Various nonlinear features are extracted from the PHOG descriptor. To balance the number of images in three classes, an adaptive synthetic sampling (ADASYN) approach is used. Two feature selection techniques namely ant colony optimization genetic algorithm (ACO-GA) and particle swarm optimization (PSO) are used to select the best performing method. The selected features are subjected to analysis of variance (ANOVA) to determine highly significant features for classification. The proposed system has achieved a maximum accuracy of 85.1%, sensitivity of 87.2%, and specificity of 80% using nine features selected by PSO feature selection method with support vector machine (SVM) classifier.

2.2 Theoretical Analysis of Glaucoma

Philadelphia Telemedicine Glaucoma Detection and Follow-up Study: confirmation between eye screening and comprehensive eye examination diagnoses

This project aims to evaluate agreement between ocular findings of a telemedicine eye screening (visit 1) with diagnoses of a comprehensive eye examination (visit 2). A primary care practice (PCP)–based telemedicine screening program incorporating fundus photography, intraocular pressure (IOP) and clinical information was conducted. Eligible individuals were African American, Hispanic/Latino or Asian over the age of 40; Caucasian individuals over age 65; and adults of any ethnicity over age 40 with a family history of glaucoma or diabetes. Participants with abnormal images or elevated IOP were invited back for a complete eye examination. Both visit 1 and visit 2 were conducted at participants' local PCP. Ocular findings at visit 1 and eye examination diagnoses at visit 2 are presented, including a cost analysis. Results Of 906 participants who attended visit 1, 536 were invited to visit 2 due to ocular findings or unreadable images. Among the 347 (64.9%) who attended visit 2, 280 (80.7%) were diagnosed with at least one ocular condition. Participants were predominately women (59.9%) and African American (65.6%), with a mean age (±SD) of 60.6±11.0 years. A high diagnostic confirmation rate

(86.0%) was found between visit 1 and visit 2 for any ocular finding. Of 183 with suspicious nerves at visit 1, 143 (78.1%) were diagnosed as glaucoma or glaucoma suspects at visit2.

Automated Detection of Glaucoma Using Image Processing TechniquesTechniques by Mishkin Khunger, Tanupriya Choudhury, Suresh Chandra Satapath and Kuo

Glaucoma is one of the most dreaded eye diseases and is a chronically progressive and is chemic optical neuropathy leading to deterioration of vision generally caused due to increased pressure caused by increasing aqueous humous inside the eye. This is caused either due to reduced drainage or sometimes due to increased secretion. It causes damage of ischemic to the optical nerve which results in nerve fiber layer damage and permanent loss of vision. Two kinds of primary Glaucoma are there, namely wide-angle glaucoma and narrow-angle glaucoma which have diverse mechanism of lessening watery surge and are in charge of increment in intraocular pressure. In the beginning of glaucoma, no detectable side effects show up. As the ailment advances, vision exacerbates and harm to visual field occurs. If undetected and untreated, it results in complete vision loss. Manual investigation of ophthalmic images is tedious, and accuracy relies upon the skill of the experts. Programmed examination of retinal pictures is turning out to be of great importance these days. It helps in detecting, diagnosing and anticipating of dangers related with glaucoma. Fundus pictures acquired from fundus camera have been utilized for the investigation. The systems specified in the present survey have certain positive and negative points. In view of this investigation, one can undoubtedly figure out which strategy gives the ideal outcome.

3.MODULES IN DETECTION OF GLAUCOMA AND ARMD

The project consists of the following modules :

1. Module for Glaucoma Detection
2. Module for ARMD Detection

3.1 Module for Glaucoma Detection :

Glaucoma is a condition that damages your eye's optic nerve. It gets worse over time. It's often linked to a buildup of pressure inside your eye. Glaucoma tends to run in families. You usually don't get it until later in life. The increased pressure in your eye, called intraocular pressure, can damage your optic nerve, which sends images to your brain. If the damage worsens, glaucoma can cause permanent vision loss or even total blindness within a few years.

D-Eye model for Glaucoma :

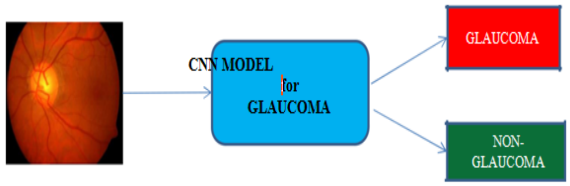


Figure : 3.1 Model for Glaucoma Detection

Figure 3.1 depicts the block diagram of model for Glaucoma Detection, the model takesfundus image as input and predicts whether the fundus is suffering from Glaucoma or not.

The fundus eye images have been taken from kaggle and the deep learning model is then trained to recognize the presence or absence of disease by prediction, with certain accuracy, when a fundus eye image is given as input.

3.2 Module for ARMD detection :

Age-related macular degeneration — also called macular degeneration, AMD or ARMD is deterioration of the macula, which is the small central area of the retina of the eye that controls visual acuity.The health of the macula determines our ability to read, recognize faces, drive, watch television, use a computer or phone, and perform any other visual task that requires us to see fine detail.

D-Eye model for ARMD

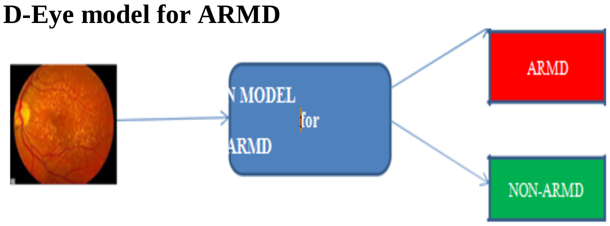


Figure : 3.2 Model for ARMD Detection

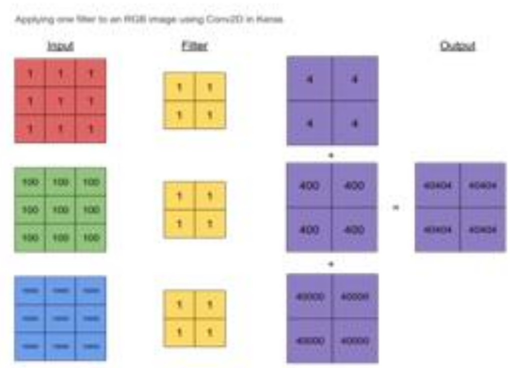
Figure 3.2 depicts the block diagram of model for Glaucoma Detection, the model takes fundus image as input and predicts whether the fundus is suffering from ARMD or not.

The fundus eye images have been taken from stare, the deep learning model is then trained to recognize the presence or absence of disease by prediction, with certain accuracy, when a fundus eye image is given as input.

4.METHDOLOGY OF DETECTION OF GLAUCOMA AND ARMD

4.1Data set Analysis :

- The Glaucoma dataset is collected from Kaggle.
- The kaggle data set consists of around 500 images belonging to Glaucoma and other 500 belonging to Non Glaucoma.
- The dataset for ARMD is taken from State Dataset.
- The stare data set consists of around 900 images belonging to ARMD disease.



4.2 Data Pre-Processing :

- Preprocessing refers to all the transformations on the raw data before it is fed to the machine learning or deep learning algorithm.
- In the preprocessing, the size of all the images are fit into frame of size, (128, 128).
- Rescaling is done to transform every pixel value from range [0,255] -> [0,1].
- shearing displaces each point in the vertical direction by an amount proportional to its distance from an edge of the image.
- This transformation zooms the initial image in or out.
- It flips the image with respect to the vertical axis. One can either turn it on or off using the horizontal_flip parameter.

4.3 Converting Images to Pixels :

- Computers see an input image as an array of pixels.

- Based on the image resolution, it will see h x w x d (h = Height, w = Width, d = Dimension).
- Eg., An image of 6 x 6 x 3 array of a matrix of RGB (3 refers to RGB values) and an image of 4 x 4 x 1 array of a matrix of grayscale image.

- tf.keras.preprocessing.image.img_to_array(img, data_format=None, dtype=None)

- It returns a 3D numpy array.

4.4 CNN Architecture of Detection of Glaucoma and ARMD :

The Artificial neural networks are the computing systema vaguely inspired by biological neural networks. Such systems learn to perform tasks by considering examples. Among various Artificial neural networks, Convolutional Neural networks(CNN) are considered efficient for image classification. CNN image classification takes an input image, processes it and classifies it under certain specified categories.

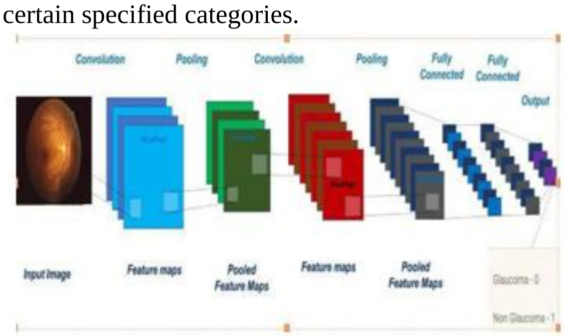


Figure 4.1 : CNN Architecture of Detection of Glaucoma and ARMD

Figure 4.1 shows the architecture of Convolutional Neural Network used for this project, consisting of input fundus image and all the convolution and pooling layers and output predictor.

Each input image will pass through a series of convolution layers. In which a filter is used to reduce the size of the image matrix as specified. Pooling is performed in order to reduce the number of trainable parameters. Subsequently, pooling layers are added between convolution layers. In the Fully Connected layer, the results of convolution/pooling layers are used to classify the image accordingly.

- CNN image classifications take an input image, process it and classifies it under certain categories.

- Each input image will pass through a series of convolution layers.
- Pooling is performed in order to reduce the number of trainable parameters.
- Subsequently, pooling layers are added between convolution layers.

- In the Fully Connected layer, the results of convolution/pooling layers are used to classify the image accordingly.

Conv2D Layer :

conv2D: to perform convolution operation a filter is used whose size can be specifed ,atlast a matrix is obatined, which is much smaller than the input matrix.

$$V_t = \gamma V_{t-1} + \eta \nabla_\theta J(\theta)$$
$$\theta = \theta - V_t$$

Figure 4.2 : Example for functioning of Convolution LayerFigure 4.2 represents how the Convolution works for a given input of matrices.

MaxPooling2Layer :

somtimes when image is too large, we would need to reduce the number of trainable parameters. subsequently we add pooling layers between convolution layers. The most common from of pooling layer generally applied is the max pooling. Pooling layers provide an approach to down sampling feature maps by summarizing the presence of features in patches of the feature map. Two common pooling methods are average pooling and max pooling that summarize the average presence of a feature and the most activated presence of a feature respectively.

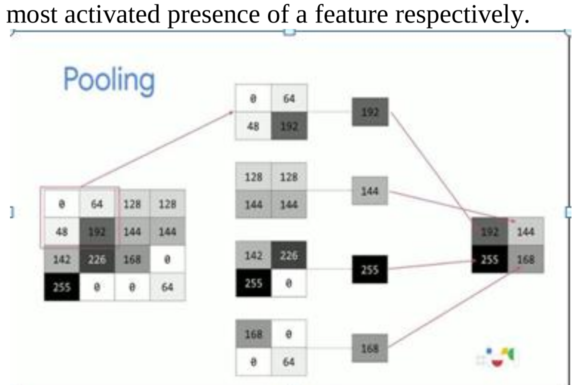


Figure 4.3 : Example for Maxpooling Layer

Figure 4.3 shows the working of MaxPooling2D layer for a given matrix as input.

4.5 Activation Functions :

ReLu Activation Function :

- ReLU stands for rectified linear unit, and is a type of activation function. Mathematically, it is defined as y = max(0, x).
- ReLU is linear (identity) for all positive values, and zero for all negative values.

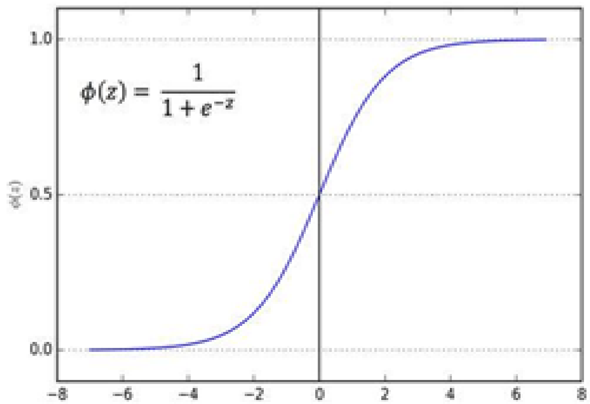


Figure 4.4 : Graph of ReLu Activation Function

Figure 4.4 is the graphical representation of ReLu Activation Function

Sigmoid Activation Function :

- Sigmoid activation function, sigmoid(x) = 1 / (1 + exp(-x))
- For small values (<-5), sigmoid returns a value close to zero, and for large values (>5) the result of the function gets close to 1

Figure 4.5 : Graph of Sigmoid Activation Function

Figure 4.5 shows the graphical representation of Sigmoid Activation Function.

4.6 Adam Optimizer Algorithm :

Adam optimization is an extension to Stochastic gradient decent and can be used in place of classical stochastic gradient descent to update network weights more efficiently.

Momentum

When explaining momentum, researchers and practitioners alike prefer to use the analogy of a ball rolling down a hill that rolls faster toward the local minima, but essentially what we must know is that the momentum algorithm, accelerates stochastic gradient descent in the relevant direction, as well as dampening oscillations.To introduce momentum into our neural network, we add a temporal element to the update vector of the past time step to the current update vector. This gives the effect of increased momentum of the ball by some amount. This can be expressed mathematically as:

Adaptive Learning Rate

Adaptive learning rates can be thought of as adjustments to the learning rate in the training phase by reducing the learning rate to a pre-defined schedule of which we see in AdaGrad, RMSprop, Adam and AdaDelta — This is also referred to as Learning Rate Schedules.

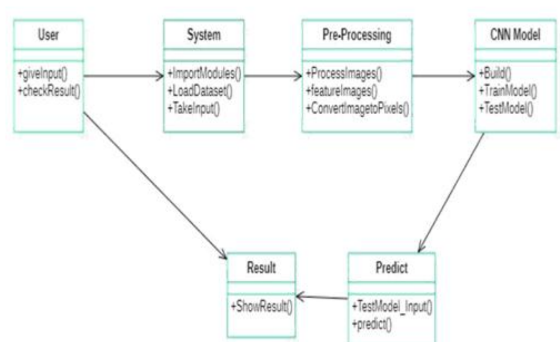


Figure 4.6 : Adam Optimization Algorithm

Figure 4.6 shows the steps involved in Adam optimization Algorithm which is used in this project for optimization of CNN models.

This project uses Adam Algorithm for optimization as it is efficient because it is a combination of Stochastic Gradient Descent with momentum and adaptive learning. So, it gives better results than rmsprop and other algorithms.

5.SYSTEM DESIGN OF DETECTION OF GLAUCOMA AND ARMD

5.1 SYSTEM ARCHITECTURE OF DETECTION GLAUCOMA AND ARMD :

The system basically uses Jupyter notebook which sets the path to read the data set. The data set is downloaded from kaggle and sent for data pre-processing . In the pre-processing the data is extracted, the images are resized and are converted to pixels.Then this converted pixels are feeded to the Deep Learning Model. After training evaluate new data with the model to get predicted results. Figure 5.1 shows the system architecture of the proposed system.

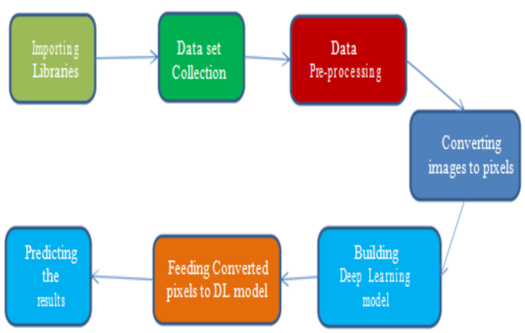


Figure 5.1 : System Architecture of Detection of Glaucoma and ARMD

Figure 5.1 shows the system architecture of Detection of Glaucoma and ARMD. It shows all the processes in this project from importing libraries to loading data, data pre-processing, converting images to pixels then building and training our Deep Learning model , training the model and testing the model.

5.2 SDLC Methodologies used for Detection of Glaucoma and ARMD

SDLC MODEL:

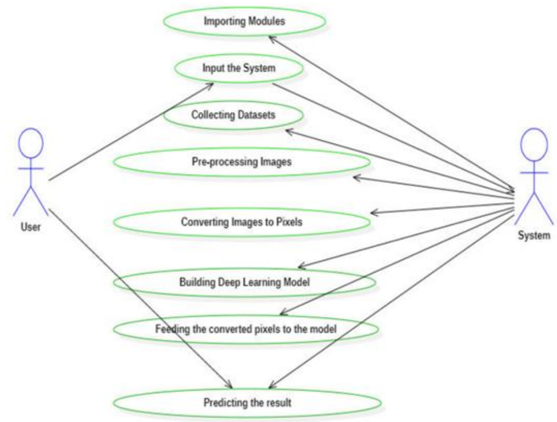


Figure 5.2 : project life-cycle of Detection of Glaucoma and ARMD

Figure 5.2 shows the general project life cycle of this project.

The Software Development Lifecycle(SDLC) for small to medium database application development efforts. This project uses iterative development lifecycle, where components of the application are developed through a series of tight iteration. The first iteration focus on very basic functionality, with subsequent iterations adding new functionality to the previous work and or correcting errors identified for the components in production.

The six stages of the SDLC are designed to build on one another, taking outputs from the previous stage, adding additional effort, and producing results that leverage the previous effort and are directly traceable to the previous stages. During each stage, additional information is gathered or developed, combined with the inputs, and used to produce the stage deliverables. It is important to not that the additional information is restricted in scope, new ideas that would take the project in directions not anticipated by the initial set of high-level requirements or features that are out-of-scope are preserved for later consideration.

Too many software development efforts go awry when development team and customer personnel get caught up in the possibilities of automation. Instead of focusing on high priority features, the team can become mired in a sea of nice to have features that are not essential to solve the problem, but in themselves are highly attractive.

5.3 UML Diagrams:

The Unified Modeling Language (UML) is used to specify, visualize, modify, construct and document the artifacts of an object-oriented software intensive system under development. UML offers a standard way to visualize a system's architectural blueprints, including elements such as:

1. Actors
2. Business processes
3. Components
4. Activities
5. Programming language statements
6. Database schemes, and Reusable software components

UML combines best techniques from data modeling (entity relationship diagrams), business modeling (work flows), object modeling, and component modeling. It can be used with all processes, throughout the software development life cycle, and across different implementation technologies. UML has synthesized the notations of the Booch method, the Object-modeling technique (OMT) and Object-oriented software engineering (OOSE) by fusing them into a single, common and widely usable modeling language. UML aims to be a standard modeling language which can model concurrent and distributed systems.

5.3.1 Class Diagram:

A class diagram is a type of diagram and part of a unified modelling language (UML) that defines and provides the overview and structure of a system in terms of classes, attributes and methods, and the relationships between different classes.

Class diagram is used to illustrate and create a functional diagram of the system classes and serves as a system development resource within the software development life cycle. Class has three main parts illustrated in rectangular boxes[5]. The first or top part specifies the class name, the second or middle specifies attributes of that class and the third or bottom section lists the methods or operations that specific class can perform.

Require: α: Stepsize
Require: $\beta_1, \beta_2 \in [0,1)$: Exponential decay rates for the moment estimates
Require: $f(\theta)$: Stochastic objective function with parameters θ
Require: θ_0: Initial parameter vector
$m_0 \leftarrow 0$ (Initialize 1st moment vector)
$v_0 \leftarrow 0$ (Initialize 2nd moment vector)
$t \leftarrow 0$ (Initialize timestep)
while θ_t not converged **do**
$t \leftarrow t + 1$
$g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$ (Get gradients w.r.t. stochastic objective at timestep t)
$m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (Update biased first moment estimate)
$v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ (Update biased second raw moment estimate)
$\hat{m}_t \leftarrow m_t / (1 - \beta_1^t)$ (Compute bias-corrected first moment estimate)
$\hat{v}_t \leftarrow v_t / (1 - \beta_2^t)$ (Compute bias-corrected second raw moment estimate)
$\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon)$ (Update parameters)
end while
return θ_t (Resulting parameters)

Figure 5.3 Class diagram for Detection of Glaucoma and ARMD

Figure 5.3 This class diagram shows the relation among the work being done in a sequence order. The modules are first imported in python notebook, datasets are loaded, and data images are being pre- processed. The deep learning model extracts the features and is trained to predict the disease.

5.3.2 Use case Diagram:

Use case Diagrams represent the functionality of the system from a user's point of view. Use cases are used during requirements elicitation and analysis to represent the functionality of the system[4]. Use cases focus on the behavior of the system from external point of view. Actors are external entities that interact with the system

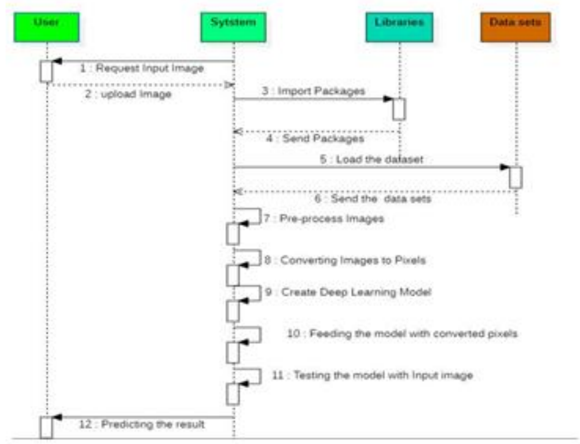


Figure 5.4: Use case diagram for Detection of Glaucoma and ARMD.

Figure 5.4 The Use case diagram above, describes the work done by the user and the System.

The series of steps collectively was performed subsequently in an order.

5.3.3 Sequence Diagram:

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios. A sequence diagram[4] shows, as parallel vertical lines, different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

5.3.4 State Diagram:

A state diagram shows the behavior of classes in response to external stimuli. Specifically a state diagram describes the behavior of a single object in response to a series of events in a system.

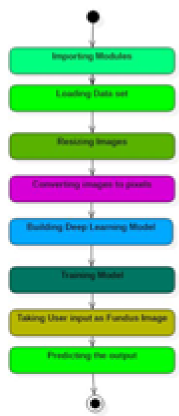


Figure 5.6 : the State Diagram of Detection of Glaucoma and ARMD

Figure 5.6 is The state diagram, that shows how the software states of the project are related and leads to a complete project at the end.

6.TESTING AND RESULTS

6.1 Testing :

Machine Learning Testing (ML testing) refers to any activity designed to reveal machine learning bugs. An ML bug may exist in the data, the learning program, or the framework.

6.1.1 Offline Testing:

At the very beginning, developers need to conduct requirement analysis to define the expectations of the users for the machine learning system under test. In requirement analysis, specifications of a machine learning system are studied and the whole testing procedure is planned. After that, test inputs are either sampled from the collected data or generated based on a specific purpose. Test oracles are then identified or generated. When the tests are ready, they need to be executed for developers to collect results[8]. The test execution process involves building a model with the tests (when the tests are training data) or running a built model against the tests (when the tests are test data), as well as checking whether the test oracles are violated[3]. After the process of test execution, developers may use evaluation metrics to check the quality of tests, i.e., the ability of the tests to expose ML problems.

The test execution results yield a bug report to help developers to duplicate, locate, and solve the bug. Those identified bugs will be labelled with different severity and assigned for different developers[8]. Once the bug is debugged and repaired, regression testing is conducted to make sure the repair solves the reported problem and does not bring new problems. If no bugs are identified, the offline testing process ends, and the model is deployed.

6.1.2 Online Testing:

Offline testing tests the model with historical data without in the real application environment. Online testing complements the shortage of offline testing, and aims to detect bugs after the model is deployed online.

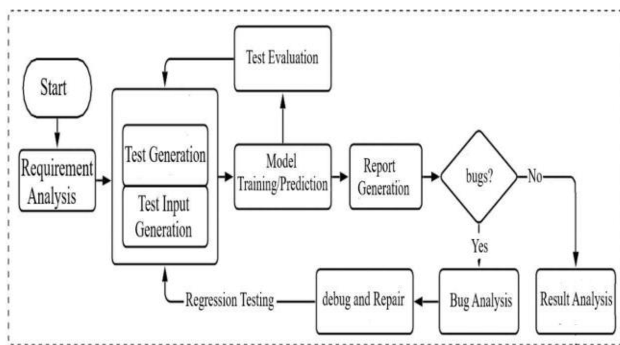


Figure 6.1 Workflow of Detection of Glaucoma and ARMD Testing

The workflow of online testing is shown in Figure 6.1. There are different methods of conducting online testing for different purposes. For example, runtime monitoring keeps checking whether the running ML systems meet the requirements or violate some desired runtime properties. Another commonly used scenario is to monitor user responses[8], based on which to find out whether the new model is superior to the old model under certain application contexts. A/B testing is one typical type of such online testing. When performing A/B testing on ML systems, the sampled users will be split into two groups using the new and old ML models separately. MAB (Multi-Armed Bandit) is another online testing approach[3]. It first conducts A/B testing for a short time and finds out the best model, then put more resources on the chosen model.

Testing Properties:

Testing properties refer to what to test in testing process and for what conditions testing needs to guarantee for a trained model. The following are some typical properties of the proposed model.

➢ Correctness: Correctness measures the probability that the proposed system under test

➢ Data Privacy: Data Privacy in machine learning is the proposed system's ability to preserve private data information

➢Efficiency: The efficiency of a machine learning system refers to its construction or prediction speed

➢Security: The security of an machine learning system is the system's resilience against potential harm, danger, or loss made via manipulating or illegally accessing ML components.

6.1.3INPUT AND OUTPUT DESIGN

The following some are the projects inputs and outputs.

INPUT:

- Importing the all required packages like numpy, pandas, matplotlib, scikit – learn and required machine learning algorithms packages.
- Setting the dimensions of visualization graph.
- Downloading and importing the dataset and convert to data frame.

OUTPUT:

- Preprocessing the importing data frame for imputing nulls with the related information.
- All are displaying cleaned outputs.
- After applying machine learning algorithms it will give good results and visualization plots.

INPUT DESIGN

Input design is a part of overall system design. The main objective during the input design is as given below:

- To produce a cost-effective method of input.
- To achieve the highest possible level of accuracy.
- To ensure that the input is acceptable and understood by the user.

OUTPUT DESIGN

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provide a permanent copy of the results for later consultation. The various types of outputs in general are:

- External Outputs, whose destination is outside the organization,
- Internal Outputs whose destination is within organization and they are the
- Operational outputs whose use is purely within the computer department.
- Interface outputs, which involve the user in communicating directly with the outputs were needed to be generated as a hard copy and as well as queries to be viewed on the screen. Keeping in view these outputs, the format for the output is taken from the outputs, which are currently being obtained after manual processing. The standard printer is to be used as output media for hard copies.

6.2 Output Screens

Figure 6.2 : The output indicating Non-Glaucoma

Figure 6.2 shows the output predicted by Glaucoma detector as Non-Glaucoma correctly on server side i.e, colab notebook for given non-Glaucoma fundus image as input.

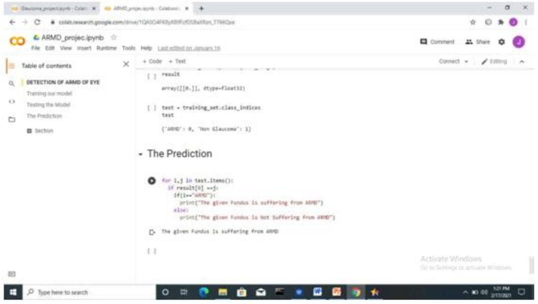


Figure 6.3 : The output indicating Glaucoma

Figure 6.3 shows the output predicted by Glaucoma Detection Model as Glaucoma for the given fundus image on colab notebook.

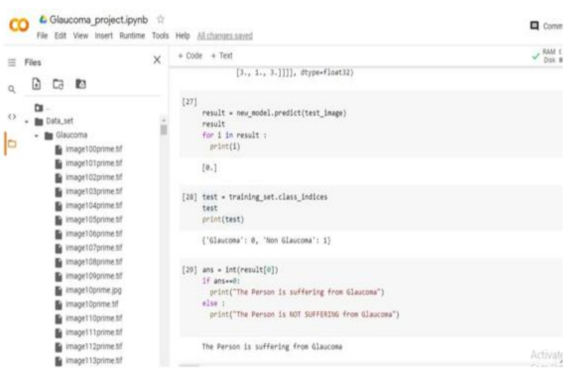


Figure 6.4 : The output for ARMD

Figure 6.4 shows the output predicted by the ARMD model for a given fundus image as input.

```
classifier.summary()
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 62, 62, 32)        896
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 31, 31, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 29, 29, 32)        9248
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 14, 14, 32)        0
_________________________________________________________________
flatten (Flatten)            (None, 6272)              0
_________________________________________________________________
dense (Dense)                (None, 128)               802944
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 129
=================================================================
Total params: 813,217
Trainable params: 813,217
Non-trainable params: 0
_________________________________________________________________
```

Figure 6.5 : The Training parameters for CNN model

Figure 6.5 shows all the trainable parameters count in each layer of the Galucoma and ARMD Detection Model and output shape of each layer in CNN architecture of this project.

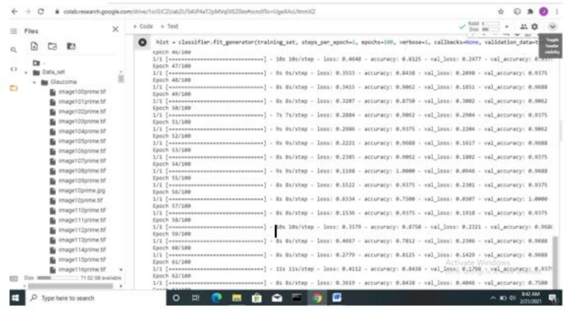


Figure 6.6 : The Epochs of Training DL Model

Figure 6.6 shows the training of Glaucoma and ARMD model for the given training dataset usally referred as epochs. In each epoch the model goes through the entire dataset once and reports accuracy and loss values.

Figure 6.7 : The User Interface for Detection of Glaucoma

Figure 6.7 shows shows the output screen of Graphical User Interface for an user to upload

his/her fundus image and check whether it is suffering from Glaucoma or not.

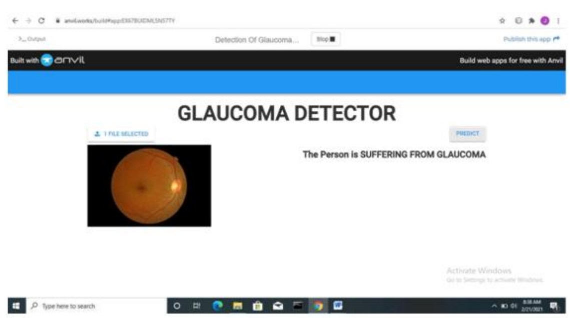


Figure 6.8 : The User Interface for Detection of ARMD

Figure 6.8 shows the output screen of Graphical User Interface for an user to upload his/her fundus image and check whether it is suffering from ARMD or not.

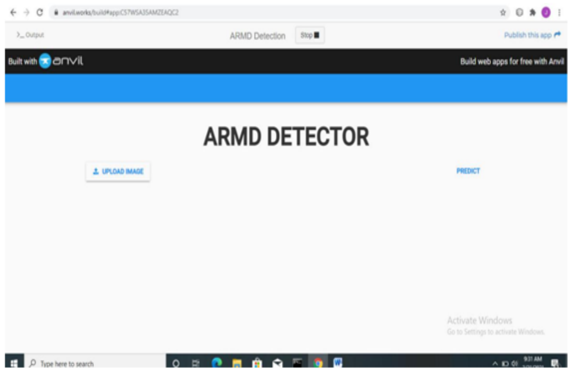


Figure 6.9 : The User Interface Output Screen indicating Glaucoma

Figure 6.9 shows the output screen of Graphical User Interface indicating that the given fundus image is suffering from Glaucoma.

Figure 6.10: The User Interface indicating NOT – Glaucoma

Figure 6.10 shows the output screen that the given fundus image is not suffering from Glaucoma.

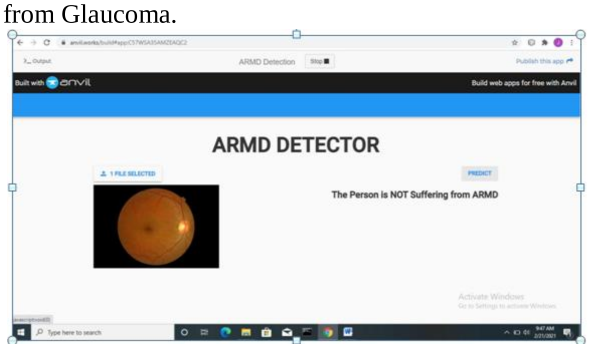


Figure 6.11 : The User Interface Output Screen indicating ARMD

Figure 6.11 shows the user interface output screen indicating that the given fundus image is suffering from ARMD.

Figure 6.12 : The User Interface Output Screen indicating NOT-ARMD

7.1 Conclusion

The Detection of Glaucoma and ARMD project is developed with a motive to help Eye care workers, to help reduce the time taken for generating a diagnosing report of disease recognition, which would manually take upto 2 weeks of time. This system can be used for mass Eye care camps across remote areas, which would hugely benefit the people living in such areas.

Though the system gives good efficiency, it is recommended to be used only by trained officials, under the care of ophthalmologists.

➢The testing efficiency of the system in recognizing the Glaucoma disease is 0.9062

➢The testing loss of the system in recognizing the Glaucoma disease is 0.1876

➢The testing efficiency of the system in recognizing the ARMD disease is 0.7164

The testing loss of the system in recognizing the Glaucoma disease is 0.2365

7.2 Future Scope

Future algorithms involving drusen detection should aim to provide useful quantification to aid screening for ARMD. A screening program should stratify patients according to optimal follow-up pathway. For automated drusen detection to contribute to the cost-effectiveness of a screening program for ARMD, it must separate individuals with drusen associated with normal aging from patients whose drusen load progresses and stratify patients with mild ARMD into those at low risk and at high risk of progression to severe ARMD. This would enable the ophthalmologist to select relevant patients for regular follow-up, thus improving the efficiency of patient care.

The Datasets available are now foreign datasets, and with the help of a device which is reliable in collection the fundus eye images with HD quality would be a future scope for the further development of the project.

BIBLIOGRAPHY

[1]Acharya U, Hagiwara Y, Koh J, Salatha. Automated screening tool for dry and wet age-related macular degeneration (ARMD) using pyramid of histogram of orientated gradients (PHOG) and nonlinear features.Comput Sci. 2017;20:41e51

[2]AREDS Group. A randomized, placebo-controlled, clinical trial of high-dose supplementation and vitamins C and E, beta carotene, and zinc for age-related macular degeneration and vision loss: AREDS report no 8. Arch Ophthalmol. 2001;119:1417e36

[3]AREDS. The age-related eye disease study severity scale for age-related macular degeneration. Arch Opthalmology. 2006;123(11):1484e98

*[4]Automated detection of age-related macular degeneration in color fundus photography: a systematic review Emma Pead, MSa, *, Roly Megaw, MDb , James Cameron, PhD FRCOpthc , Alan Fleming, PhDd , Baljean Dhillon, FRCsEDb , Emanuele Trucco, PhDe,y , Thomas MacGillivray, PhDa,y*

[5]Automated Detection of Glaucoma Using Image Processing Techniques Mishkin Khunger, Tanupriya Choudhury, Suresh Chandra Satapathy and Kuo-Chang Ting

[6]Evaluation of a deep learning system for the joint automated detection of diabetic retinopathy and age-related macular degeneration Cristina Gonzalez-Gonzalo,1,2,3,4 Veronica S anchez-Gutierrez,5 Paula Hernandez-Martınez,5 Ines Contreras,5,6 Yara T. Lechanteur,4 Artin

[7]. Bennilova I, Karran E, De Strooper B. The toxic Ab oligomer and Alzheimer's disease: an emperor in need of clothes. Nat Neurosci. 2012;15(3):349e5

[8]Brandi C, Breinlich V, Stark KJ, Enzinger S. Features of AgeRelated Macular Degeneration in the General Adults and Their Dependency on Age, Sex, and Smoking: Results from the German KORA Study. PLoS One. 2016;11:e0167181

[9]Burlina P, Freund D, Dupas B, Bressler N. Automatic screening of Age-related macular degeneration and retinal abnormalities. 33rd Annual International Conference .

[10]Kesar T. N, T. C Manjunath, 'Diagnosis & detection of eye diseases using Deep Convolutional Neural Networks & Raspberry Pi', Second IEEE International Conference on Green Computing & Internet of Things (IOT), ICGCIoT, IEEE ISBN: 978-1-5386-5657-0

https://www.kaggle.com/himanshuagarwal1998/glaucoma

[11]Philadelphia Telemedicine Glaucoma Detection and Follow-up Study: confirmation between eye screening and comprehensive eye examination diagnoses

[12]SEMI-SUPERVISED TRANSFER LEARNING FOR CONVOLUTIONAL NEURAL NETWORKS FOR GLAUCOMA DETECTION Manal Al Ghamdi Mingqi Li† Mohamed Abdel-Mottaleb‡ Mohamed Abou Shousha