**DIABETIC RETINOPATHY-OPTICAL LOSS DETECTION USING OCT IMAGE THROUGH DEEP LEARNING**

## A PROJECT REPORT

***Submitted by***

# BALAJI R [211418104034]

**GOKUL N [211418104067]**

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

***of***

# BACHELOR OF ENGINEERING

**IN**

# COMPUTER SCIENCE AND ENGINEERING



## PANIMALAR ENGINEERING COLLEGE

**(An Autonomous Institution, Affiliated to Anna University, Chennai)**

## MAY 2022

**PANIMALAR ENGINEERING COLLEGE**

**(An Autonomous Institution, Affiliated to Anna University, Chennai)**

# BONAFIDE CERTIFICATE

Certified that this project report **“DIABETIC RETINOPATHY-OPTICAL LOSS DETECTION USING OCT IMAGE THROUGH DEEP LEARNING”** is the bonafide work of “**BALAJI R (211418104034), GOKUL N (211418104067)**” who carried out the project work under **Mrs. K.SANGEETHA, M.E,** supervision.

## 

## SIGNATURE SIGNATURE

**Dr. S.MURUGAVALLI, M.E, Ph.D.., Mrs. K.SANGEETHA, M.E.,**

**HEAD OF THE DEPARTMENT SUPERVISOR**

## 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 mentioned students were examined in End Semester Project Viva-Voice Examination held on ...........................

## INTERNAL EXAMINER EXTERNAL EXAMINER

**DECLARATION BY THE STUDENT**

## We BALAJI R (211418104034), GOKUL N(211418104067),hereby declare that this project report titled “DIABETIC RETINOPATHY-OPTICAL LOSS DETECTION USING OCT IMAGE THROUGH DEEP LEARNING”, under the guidance of Mrs.K.SANGEETHA, M.E., is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

## BALAJI.R

* 1. **GOKUL.N**

**ACKNOWLEDGEMENT**

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**, **Dr.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 our **Project Guide Mrs. K. SANGEETHA, M.E.,** and all the faculty members of the Department of CSE for their advice and suggestions for the successful completion of the project.

Lastly, we take the opportunity to thank all our staff members of the department of computer science and engineering. Regards to our family and friends who offered an unflinching moral support for the completion of this project.

## BALAJI R (211418104034)

**GOKUL N (211418104067)**

**ABSTRACT**

Diabetes is a disease that occurs when there is an excess level of glucose content in body that may cause damage to retina which leads to permanent loss of vision in stages .Diabetes affecting the eyes is known as Diabetes retinopathy

,which become global medical issue among elderly people .Diabetes retinopathy is a mellitus disease that affects blood vessels in retina .This requires an immediate treatment to keep remaining vision from damage .Here comes the need for automatic system to detect possibility of people getting affected with diabetic retinopathy at initial stage. The main objective of the model is to classify normal and abnormal retinal images into four stages of DR with help of classification algorithms and to apply Deep learning (DL) model to the field of medical diagnosis in order to lessen the time and stress undergone by the ophthalmologist and other members of team who are involved in the process of screening, diagnosis and treatment of diabetic retinopathy .The optical coherence tomography (OCT) images of diabetic retinopathy dataset is used for training which is available for free in the internet. Later the validation and evaluation are done using specific deep learning technique. The entire proposed project has a simple user-friendly User interface (UI) which makes the user to identify the type of Diabetic retinopathy easily. Once the test image is uploaded in the system, and the user interface will have some buttons in order to do the necessary operations on the given image and prediction processes and displays the output.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
|  | ABSTRACT | **v** |
|  | LIST OF TABLES | **viii** |
|  | LIST OF FIGURES | **ix** |
|  | LIST OF ACRONYMS & ABBREVIATIONS | **x** |
| **1.** | **INTRODUCTION** | **1** |
|  | 1.1 INTRODUCTION | **2** |
|  | 1.2 PROBLEM STATEMENT | **4** |
| **2.** | **LITERATURE SURVEY** | **6** |
| **3.** | **SYSTEM ANALYSIS** | **13** |
|  | 3.1 EXISTING SYSTEM | **14** |
|  | 3.2 PROPOSED SYSTEM | **15** |
|  | 3.3 FEASIBILITY STUDY | **15** |
|  | 3.3.1 INTRODUCTION | **15** |
|  | 3.3.2 FINANCIAL FEASIBILITY | **16** |
|  | 3.3.3 TECHNICAL FEASIBILITY | **16** |
|  | 3.3.4 RESOURCE FEASIBILITY | **16** |
|  | 3.4 HARDWARE ENVIRONMENT | **17** |
|  | 3.5 SOFTWARE ENVIRONMENT | **17** |
| **4.** | **SYSTEM DESIGN** | **18** |
|  | 4.1 USE-CASE DIAGRAM | **19** |
|  | 4.2 ACTIVITY DIAGRAM | **20** |
|  | 4.3 SEQUENCE DIAGRAM | **21** |
|  | 4.4 DATA FLOW DIAGRAM | **21** |
|  | 4.4.1 DATA FLOW DIAGRAM 0 | **24** |
|  | 4.4.2 DATA FLOW DIAGRAM 1 | **25** |
|  | 4.5 ER DIAGRAM | **25** |
| **5.** | **SYSTEM ARCHITECTURE** | **27** |
|  | 5.1 SYSTEM ARCHITECTURE | **28** |
|  | 5.2 MODULE DESCRIPTION | **28** |
|  | 5.2.1 MODEL FEATURES | **28** |
|  | 5.2.2 IMAGE PROCESSING | **29** |
|  | 5.2.3 NETWORK SELECTION | **30** |
|  | 5.3 ALGORITHM | **31** |
| **6.** | **SYSTEM IMPLEMENTATION** | **32** |
|  | 6.1 INTRODUCTION | **33** |
|  | 6.2 DATA COLLECTION | **33** |
|  | 6.3 DATA PREPARATION | **33** |
|  | 6.3.1 DATA AUGMENTATION | **33** |
|  | 6.3.2 DATA CLEANSING | **34** |
|  | 6.4 TRAINING MODEL | **34** |
|  | 6.5 VGG16-FEATURE EXTRACTOR | **35** |
|  | 6.6 CLIENT SIDE CODING | **36** |
|  | 6.7 SERVER SIDE CODING | **40** |
| **7.** | **TESTING** | **48** |
|  | 7.1 SYSTEM TESTING | **49** |
|  | 7.2 TESTING TECHNIQUES | **52** |
|  | 7.3 SOFTWARE TESTING STRATEGIES | **54** |
|  | 7.4 TEST CASES | **57** |
| **8.** | **CONCLUSION** | **58** |
|  | **APPENDIX 1** | **60** |
|  | **REFERENCES** | **63** |
|  |  |  |

# LIST OF TABLES

|  |  |  |
| --- | --- | --- |
| **TABLE NO** | **TITLE** | **PAGE NO** |
| 7.1 | Test case | 57 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **TITLE** | **PAGE NO** |
| 4.1 | USE-CASE DIAGRAM | **19** |
| 4.2 | ACTIVITY DIAGRAM | **20** |
| 4.3 | SEQUENCE DIAGRAM | **21** |
| 4.4 | DATA FLOW DIAGRAM 0 | **24** |
| 4.5 | DATA FLOW DIAGRAM 1 | **25** |
| 4.6 | ER DIAGRAM | **26** |
| 5.1 | SYSTEM ARCHITECTURE | **28** |
| 6.1 | DATA AUGUMENTATION | **34** |
| 9.1 | NORMAL IMAGE DETECTED | **60** |
| 9.2 | IMAGE UPLOADING UI | **60** |
| 9.3 | HOME PAGE | **61** |
| 9.4 | DIABETIC MACULAR EDEMA DETECTED | **61** |
| 9.5 | CHOROIDAL NEOVASCULARIZATION  DETECTED | **62** |
| 9.6 | DRUSEN DETECTED | **62** |

# LIST OF ACRONYMS AND ABBREVIATIONS

DR - Diabetic Retinopathy

ML - Machine Learning

NPDR - Non-proliferative diabetic retinopathy PDR - Proliferative diabetic retinopathy

# CHAPTER 1

**INTRODUCTION**

## INTRODUCTION

Diabetes is one of the most dreadful diseases of the world. Most of the aged people are getting affected by diabetes mellitus. The rise of sugar content in the body leads to many dreadful disorders. It also decreases the natural strength of the body. The other diseases caused due to the diabetes are severe. One such disorder caused by diabetes mellitus is called as Diabetic Retinopathy. The high sugar level leads to diabetic retinopathy. It is a disorder where the eye sight reduces stage by stage and at one stage the entire eye sight goes off. Diabetic retinopathy is an eye condition that occurs due to diabetes. It can arise as a result of the high blood sugar levels that diabetes causes. Over time, having too much sugar in the blood can damage blood vessels throughout the body, including in the retina. The retina is the membrane covering the back of the eye. It detects light and sends signals to the brain through the optic nerve. If sugar blocks the tiny blood vessels that go into the retina, it can cause them to leak or bleed. The eye may then grow new blood vessels that are 2 weaker and leak or bleed more easily. If the eye starts to grow new blood vessels, this is known as proliferative diabetic retinopathy, which experts consider a more advanced stage. The early stage is known as non-proliferative diabetic retinopathy. The eye may accumulate fluid during long periods of high blood sugar. This fluid accumulation changes the shape and curve of the lens, causing changes in vision. Once a person gets their blood sugar levels under control, the lens will usually return to its original shape, and vision will improve.

## CAUSES OF DIABETIC RETINOPATHY

Diabetic retinopathy is caused by high blood sugar due to diabetes. Over time, having too much sugar in your blood can damage your retina — the part of your eye that detects light and sends signals to your brain through a nerve in the back of your eye (optic nerve). Diabetes damages blood vessels all over the body. The damage to your eyes starts when sugar blocks the tiny blood vessels that go

to your retina, causing them to leak fluid or bleed. To make up for these blocked blood vessels, your eyes then grow new blood vessels that don’t work well. These new blood vessels can leak or bleed easily.

## TYPES OF DIABETIC RETINOPATHY

People suffer from two kinds of retinopathy caused by the severity of the sugar level in the body. The two types are: • Early Retinopathy • Advanced Retinopathy

1. Early Retinopathy In this more common form — called non-proliferative diabetic retinopathy (NPDR) — new blood vessels aren't growing (proliferating). When you have NPDR, the walls of the blood vessels in your retina weaken. Tiny bulges (micro aneurysms) protrude from the vessel walls of the smaller vessels, sometimes leaking 3 fluid and blood into the retina. Larger retinal vessels can begin to dilate and become irregular in diameter, as well. NPDR can progress from mild to severe, as more blood vessels become blocked. Nerve fibers in the retina may begin to swell. Sometimes the central part of the retina (macula) begins to swell (macular edema), a condition that requires treatment.
2. Advanced Retinopathy Diabetic retinopathy can progress to this more severe type, known as proliferative diabetic retinopathy. In this type, damaged blood vessels close off, causing the growth of new, abnormal blood vessels in the retina, and can leak into the clear, jelly-like substance that fills the center of your eye (vitreous). Eventually, scar tissue stimulated by the growth of new blood vessels may cause the retina to detach from the back of your eye. If the new blood vessels interfere with the normal flow of fluid out of the eye, pressure may build up in the eyeball. This can damage the nerve that carries images from your eye to your brain (optic nerve), resulting in glaucoma.

## COMPLICATIONS

Diabetic retinopathy involves the abnormal growth of blood vessels in the retina. Complications can lead to serious vision problems: Vitreous hemorrhage. The new blood vessels may bleed into the clear, jelly-like substance that fills the center of your eye. If the amount of bleeding is small, you might see only a few dark spots (floaters). In more-severe cases, blood can fill the vitreous cavity and completely block your vision. Vitreous hemorrhage by itself usually doesn't cause permanent vision loss. The blood often clears from the eye within a few weeks or months. Unless your retina is damaged, your vision may return to its previous clarity. Retinal detachment. The abnormal blood vessels associated with diabetic retinopathy stimulate the growth of scar tissue, which can pull the retina away from the back of the eye. This may cause spots floating in your vision, flashes of light or severe vision loss. Glaucoma. New blood vessels may grow in the front part of your eye and interfere with the normal flow of fluid out of the eye, causing pressure in the eye to build up (glaucoma). This pressure can damage the nerve that carries images from your eye to your brain (optic nerve). Blindness. Eventually, diabetic retinopathy, glaucoma or both can lead to complete vision loss.

## Problem Statement

Diagnosis of DR is most commonly done by dilated eye examination that is performed by ophthalmologists. Other methods of disease diagnosis include fluorescein angiography, optical coherence tomography (OCT) or fundus photography. For fluorescein angiography, the blood flow and vascular abnormalities are photographed upon the intravenous injection of contrast dye. In OCT, the retinal structure, thickness, and edema (i.e., retinal swelling) are evaluated. Currently, diagnosis of DR is subjective and needs to be performed by a retina specialist that passed a specialized training for diagnosis and grading as the visual assessment and manual measurements of changes in retinal

vasculature and layers are deemed very complex tasks. Unfortunately, a lot of diabetic patients attempt to visit a retina specialist only with symptomatic vision loss, when their pathology gets advanced and mostly irreversible, due to inadequate access to trained eye-care professionals and tertiary eye-care services. Based on this, there is a clinically significant motivation to have an objective and non-invasive diagnostic system that is capable of not only accurately detecting DR at an early stage but also grading it. Available physical tests to detect diabetic retinopathy includes pupil dilation, visual acuity test, optical coherence tomography, etc. But they are time consuming and patients need to suffer a lot. This paper focuses on automated computer aided detection of diabetic retinopathy using deep learning model.

## Objective Of the Project

Diabetic retinopathy is a passive disease caused by diabetes mellitus. It has many severe effects on the human eye, which on not taken care of could lead to loss of eye sightedness. Thought it has several surgeries to cure and replace the eye after infection. This model makes things easier by avoiding the spread by identifying the infection as early as possible and by taking required medication. The model has a strong algorithm to find out the stages and infection exactly. Thus, the model helps in identifying and warning about the level of infection in the patient. This model is a user-friendly model where the deep learning algorithm runs internally by the trained data set to identify the level of infection in the eye. The goal here is to scale their efforts through technology; to gain the ability to automatically screen images for disease and provide information on how severe the condition may be. We shall be achieving this by building a convolutional neural network model that can automatically look at a patient’s eye image and estimate the severity of blindness in the patient. This process of automation can reduce a lot of time thereby screening the process of treating diabetic retinopathy at a large scale.

# CHAPTER 2 LITERATURE REVIEW

## TITLE 1:

"Detection of DR using Machine Learning" in IRJET, 2020.

## AUTHORS:

Aryan Kokane, Gourhari Sharma, Akash Raina, Shubham Narole, Prof. Pramila

M. Chawan,

## DESCRIPTION:

The goal of this proposed paper is to conduct a survey of different literatures where studies on Diabetic Retinopathy (DR) are done and different ML techniques are used in order to detect DR in retina. DR is an eye disease in humans with untreated diabetes which is in excess amount that may cause temporary or even permanent vision loss. Therefore, with naked eyes it becomes critical for detecting DR in the early possible phase in order to avoid blindness in humans. The aim of the authors is to identify the presence of DR by applying ML classification algorithms. Hence, the authors try and summarize the various models and techniques used along with methodologies used by them and to analyze the accuracies and results produced by their models.

## TITLE 2:

DR detection through DL techniques: A review" in Informatics in Medicine Unlocked, 2020.

## AUTHORS:

Wejdan L.Alyoubi, Wafaa M.Shalash, Maysoon F.Abulkhair

## DESCRIPTION:

This proposed paper uses CNN that are more widely adopted as a DL method in detection of Diabetic retinopathy (DR), the most recent state-of-the-art methods of detection and classification of Diabetic retinopathy fundus images using deep learning techniques are reviewed and analyzed in this paper. In additional to this, the available datasets for the color fundus retinal Diabetic retinopathy is

reviewed in this paper. Various challenging issues that require a lot more investigations were also discussed in this paper.

## TITLE 3:

"Diagnosis of the DR using the Fundus Image dataset Using deep convolutional neural network " in Mobile information systems, 2019.

## AUTHORS:

Yung-Hui Li, Nai-Ning Yeh, and Yu-Chien Chung

## DESCRIPTION:

In the paper, a novel algorithm based on deep convolutional neural network (DCNN). Unlike the traditional DCNN way, they replaced the more commonly used max- pooling layer with that of fractional max-pooling: a notable change in their model. Two of the DCNN that have a varying number of layers were trained to derive more features for the classification process. After the combination of features from metadata of the images and DCNN, they have trained a support vector machine (SVM) classifier in order to learn the underlying boundary of distribution of every class.

## TITLE 4:

"Classifying of DR Images by Using DL Model" in International Journal of Grid and Distributed Computing, 2018.

## AUTHORS:

Suvajit Dutta,, Syed Muzamil Basha, and N. Ch. S. N. Iyengar

## DESCRIPTION:

The main concept behind this proposed paper is to propose an automated model that identifies the key precedents of the DR. This Proposed Model is trained

with three types of models namely, back propagation NN, Deep Neural Network (DNN) and Convolutional Neural Network (CNN) after testing models with CPU trained Neural network gives lowest accuracy because of one hidden layer whereas the deep learning models are out performing NN. The DL models were capable in process of quantifying the features as blood vessels, fluid Drips, Exudates, Blood clots and micro aneurysms into various classes. This proposed model is able to calculate the weights which give the severity level of the patient's retina or so-called eye.

## TITLE 5:

"Classifying Diabetic Retinopathy using Deep Learning Architecture" in International Journal of Engineering Research & Technology (IJERT), 2016.

## AUTHORS:

T Chandrakumar, R Kathirvel,

## DESCRIPTION:

A proposed deep learning approach such as Deep Convolutional Neural Network (DCNN) gives high accuracy in classification of these diseases through spatial analysis. A Deep Convolutional Neural Network is more critical architecture that is inferred lot more from human visual perceptions. Among the other supervised learning algorithms used, this proposed system is designed to find a better and efficient ways to classify the fundus image input given by user with literally small pre-processing techniques. Their proposed model deployed with the dropout layer technique that has yielded about 94 percent accuracy. And additionally, their system is tested with popular databases such as STARE. The Kaggle fundus images datasets are available for free on internet.

## TITLE 6:

DR using Morphological operations and ML‖, IEEE International Advance Computing Conference (IACC), (2015).

## AUTHORS:

J.Lachure, A.V.Deorankar, S.Lachure, S.Gupta, R.Jadhav

## DESCRIPTION:

They developed a system, for the detection of red and bright lesions they used dataset fundus photographs. Micro-aneurysms are the first clinical sign of DR and it appear small red dots on retinal fundus images. In order to detect retinal micro-aneurysms, the fundus images of retina were taken from Messidor dataset. After pre-processing of images, morphological operations were done in order to identify micro-aneurysms and then features of input are extracted such as GLCM and Structural features for the classification of image into their respective type. In order for the process of classifying between normal and DR images, several classes should be represented using relevant and remarkable features. SVM classifier gave better performance.

## TITLE 7:

SVM and Neural Network based Diagnosis of Diabetic Retinopathy‖, International Journal of computer Application

## AUTHORS:

R.Priya, P.Aruna

## DESCRIPTION:

Two groups were described in this model by authors, namely Non proliferative diabetic retinopathy (NPDR) and proliferative diabetic retinopathy (PDR). In this proposed model, in order to detect and classify diabetic retinopathy, two models namely Probabilistic Neural network (PNN) and Support vector machine (SVM) were depicted and their performances were compared. After completion, the experimental results shows that PNN has an accuracy of about 89% and SVM has an accuracy of about 97%. This means that the SVM model outperforms other models utilized earlier.

## TITLE 8:

Identifying Abnormalities in the Retinal Images using SVM Classifiers‖, International Journal of Computer Applications (0975- 8887), Volume 111 – No.6, (2015).

## AUTHORS:

S.Giraddi, J Pujari, S.Seeri

## DESCRIPTION:

The aim of this proposed paper is that to develop and validation of systems for the process of detection of hard exudates of retina and then to classify the input image as normal or infected ones. The authors of this paper proposed a model and implemented a novel method which is based on color and texture features of the input image. Analysis of SVM and KNN classifiers performance measure is presented. Images classified by these classifiers are validated by expert ophthalmologists.

## TITLE 9:

Transformed Representations for CNN in Diabetic Retinopathy Screening‖, Modern AI for Health Analytic Papers from the AAAI (2014).

## AUTHORS:

G.Lim, M.L.Lee, Wynne Hsu

## DESCRIPTION:

The authors of this paper demonstrate the functionality through the process of segmentation of the input fundus images with a fast, robust but loose segmentation, to obtain a set of important objects for classifying the DR. Those extracted objects later undergo a series of spatial transformations and get converted into a reduced space. Additional attributes may be abstracted as raw features that are incorporated after the convolutional phase of the network. At last, the authors compare the performance of their model against the already existing approaches/models with the challenging problems of detecting lesions

or bulges in retinal fundus images.

## TITLE 10:

Classifying Algorithm of Retinal Images of Diabetes patients based on Exudates Detection‖, 978-1-4673-2362-8/12, IEEE (2012)

## AUTHORS:

Vesna Zeljkovic, Milena Bojic, Claude Tameze; Ventzeslav Valev

## DESCRIPTION:

Automatic exudates detection and retina images classification would be helpful for reducing diabetic retinopathy screening costs and encouraging regular examinations. The authors have proposed the automated algo that applies mathematical models which makes light intensity level emphasis,by default easier exudates detection, efficient and right classification of retina fundus input images. This proposed algorithm is fast and robust to various visual changes of retinal fundus images which are commonly processed in clinics.

# CHAPTER 3 SYSTEM ANALYSIS

## EXISTING SYSTEM Neural Networks

The existing systems are based on Neural Networks which make them slow and prone to lags. The accuracy of the models is less, around 62% which makes them unreliable for real-time implementation. The existing system also draws huge computational power which makes them expensive and difficult to maintain. The existing system works on outdated ml algorithms.

## Drawbacks

* + - The existing system has less accuracy.
    - The existing system has consumed more power so maintaining is difficult.

## Machine Learning

This system makes use of the machine learning features to identify the severity of the infection. It also finds the exudates, hemorrhages, and microaneurysms by using a hybrid classifier which is a combination of random forest, k nearest neighbor, etc. Moreover, this existing system uses dataset of fundus images which requires a lot of preprocessing effort that consumes some extra time and less accurate.

## Drawbacks

* + - The resulting image will let us know whether the input image is affected by DR or not. But, the type of that infection could not be identified.
    - The reason for the infection cannot be identified using the model and doesn’t tell the value of infection also.

## PROPOSED SYSTEM

The proposed system is based on Convolutional Neural Networks, which make them fast and less prone to lags. Our proposed system makes use of infected OCT images of retina which are high resolution, contactless and non-destructive testing properties which make our model reduce some time for image processing and results in high accuracy. The accuracy of the models are comparatively high, around 95% which makes them reliable for real-time implementation. The proposed system is completely power efficient and also cost efficient and also maintenance is easy.

## Advantages of proposed system

* The proposed system is using a CNN, so it's fast and lag free.
* The proposed system has more accuracy better than existing system.
* The proposed system is in cost efficient manner.
* Proposed system makes use of CNN model which can handle large datasets of image and prediction could be done effectively.

## FEASIBILITY STUDY:

* + 1. **Introduction:**

A Feasibility Study includes a heart analysis of the necessity, value, and customary sense of a planned enterprise, like framework development. The procedure of outlining associated capital punishment record-keeping frameworks has adequate responsibility and plus suggestions for an association. the chance study can alter you to choose education and simple selection at imperative focuses throughout the biological process procedure to determine if it's operationally, economically, and indeed cheap to deliver with a selected strategy.

## Financial Feasibility:

The resources required by the proposed system are available easily in the internet for free. This procedure is to determine the benefits and savings that is expected from a system and compare with cost. Otherwise, further justification or alterations in a proposed system that have to be made if it is having a change of being approved. This is an ongoing effort that improves inaccuracy of each phase of the system lifecycle. From these it’s clear that this project is financially feasible.

## Technical feasibility:

This project is desktop-based application. The main technologies that are associated with project are

* + - * Deep Learning Model-VGG16
      * FLASK for creating web UI
      * Python for libraries
      * Visual studio code

Each of the technologies are freely on the internet and also the technical skills needed are manageable. This requires less information measure to transfer knowledge that holds on in information. From this, it's clear that our project is technically feasible.

## Resource Feasibility:

Relates to whether the participants will be able to handle the new system.

Moreover, 3 - 5 is enough for this proposed system. Resources that are required for our project are:

* + - * Programming Devices
      * Programming Tools such as Deep Learning Framework – VGG16(CNN model), VS code, Python, Flask
      * Programming Individuals who have required technical knowledge.

So, it’s clear that project has required resource feasibility.

## HARDWARE ENVIRONMENT

Processor : Pentium Dual Core 2.00GHZ

Hard disk : 120 GB

RAM : 2GB (minimum)

Keyboard : 110 keys enhanced

## SOFTWARE ENVIRONMENT

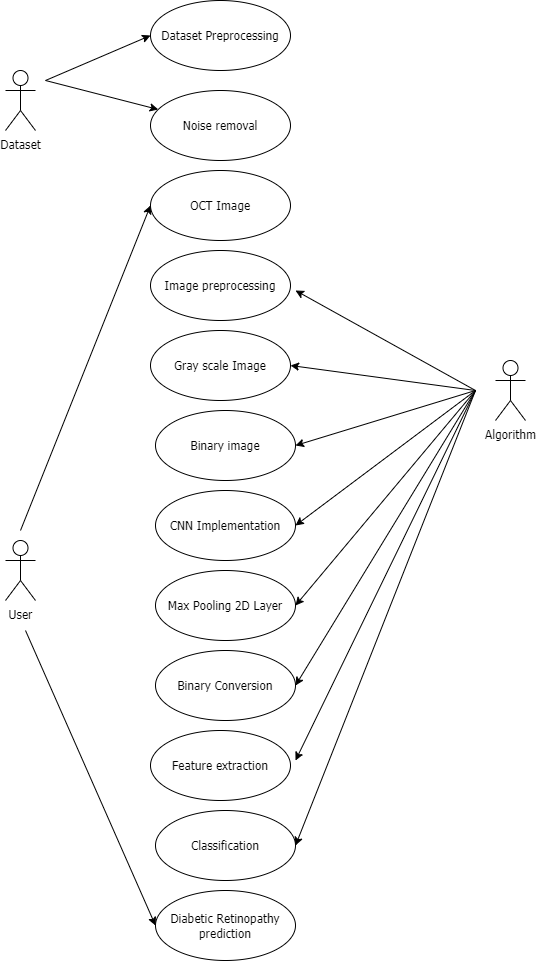
Operating system : Windows7 (with service pack 1), 8, 8.1 and 10 IDE : Visual studio code

Language : Python

# CHAPTER 4 SYSTEM DESIGN

* 1. **USE-CASE DIAGRAM**

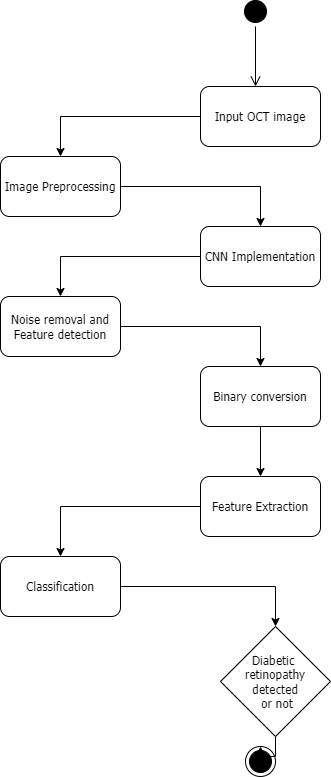
A use case diagram is a standard diagram that shows all interactions between the user, dataset, and algorithm used. It is developed in the early stages of the process.



**Fig 4.1 – Use-Case Diagram**

# ACTIVITY DIAGRAM

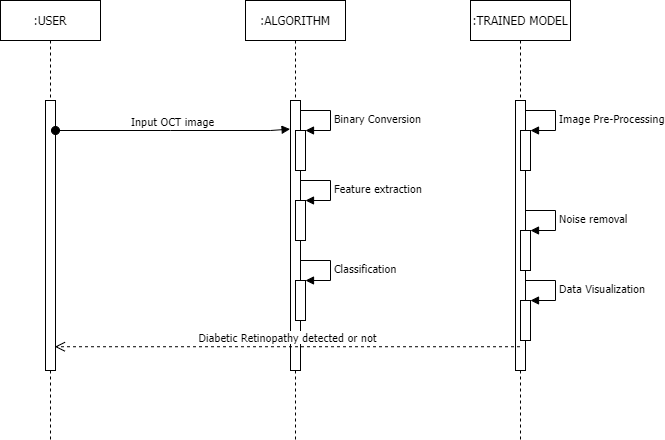
In simple terms, a diagram that represents the order of all activities is called the activity diagram. It shows the workflow between different activities that take place in the whole process. However, these are not exactly flowcharts but are similar.



**Fig 4.2 – Activity Diagram**

# SEQUENCE DIAGRAM

These are other kinds of interaction-based diagrams that show how all the operations are carried out. They capture the context of collaborations between objects and processes.



**Fig 4.3 – Sequence Diagram**

# DATA FLOW DIAGRAM

In the DFD, there are four symbols

1. A square defines a source (originating) or destination of system data
2. An arrow identifies data flow. It is the pipeline through which the information flows
3. A circle or a bubble represents a process that transforms the incoming data flow into outgoing data flows.
4. An open rectangle is a data store, data at rest or a temporary repository of data

## CONSTRUCTING A DFD:

Several rules of thumb are used in drawing DFD’S:

1. Process should be named and numbered for an easy reference. Each name should be representative of the process.
2. The direction of flow is from top to bottom and from left to right. Data traditionally flow from source to the destination although they may flow back to the source. One way to indicate this is to draw the long flow line back to a source. An alternative way is to repeat the source symbol as a destination. Since it is used more than once in the DFD it is marked with a short diagonal.
3. When a process is exploded into lower-level details, they are numbered.
4. The names of data stores and destinations are written in capital letters. Process and dataflow names have the first letter of each work capitalized

A DFD typically shows the minimum contents of data store. Each data store should contain all the data elements that flow in and out.

Questionnaires should contain all the data elements that flow in and out. Missing interfaces redundancies and like is then accounted for often through interviews.

## SAILENT FEATURES OF DFD’S

1. The DFD shows the flow of data, not of control loops and decision are controlled considerations do not appear on a DFD.
2. The DFD does not indicate the time factor involved in any process, whether the dataflow take place daily, weekly, monthly or yearly.
3. The sequence of events is not brought out on the DFD.

## TYPES OF DATA FLOW DIAGRAMS

* 1. Current Physical
  2. Current Logical
  3. New Logical
  4. New Physical

## CURRENT PHYSICAL

In Current Physical DFD process label includes the name of the people or

their positions or the names of computer systems that might provide some of the overall system-processing label includes an identification of the technology used to process the data. Similarly, data flows and data stores are often labelled with the names of the actual physical media on which data are stored such as file folders, computer files, business forms or computer tapes.

## CURRENT LOGICAL:

The physical aspects in the system are removed as much as possible so that the current system is reduced to its essence to the data and the processes that transform them regardless of actual physical form.

## NEW LOGICAL:

This is exactly like a current logical model if the user were completely happy with the user were completely happy with the functionality of the current system, but had problems with how it was implemented typically through the new logical model will differ from the current logical model while having additional functions, absolute function removal and inefficient flows recognized.

## NEW PHYSICAL:

The new physical represents only the physical implementation of the new system.

## RULES GOVERNING THE DFD’S PROCESS

1. No process can have only outputs.
2. No process can have only inputs. If an object has only inputs than it must be a sink.
3. A process has a verb phrase label.

## DATA STORE

1. Data cannot move directly from one data store to another data store, a process must move data.
2. Data cannot move directly from an outside source to a data store, a process, which receives, must move data from the source and place the data into the data store
3. A data store has a noun phrase label.

## SOURCE OR SINK

The origin and /or destination of data

1. Data cannot move direly from a source to sink it must be moved from a process
2. A source and /or sink have a noun phrase land

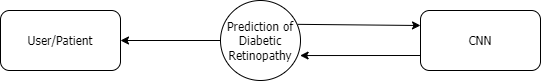
## DATA FLOW

1. A Data Flow has only one direction of flow between symbols. It may flow in both directions between a process and a data store to show a read before an update. The latter is usually indicated, however by two separate arrows since these happen at different type.
2. A join in DFD means that exactly the same data comes from any of two or more different processes data store or sink to a common location.
3. A data flow cannot go directly back to the same process it leads. There must be at least one other process that handles the data flow produce some other data flow returns the original data in the beginning process.
4. A Data flow to a data store means update (delete or change).
5. A data Flow from a data store means retrieve or use.

A data flow has a noun phrase label more than one data flow noun phrase can appear on a single arrow as long as all of the flows on the same arrow move together as one package.

# DATA FLOW DIAGRAM LEVEL 0

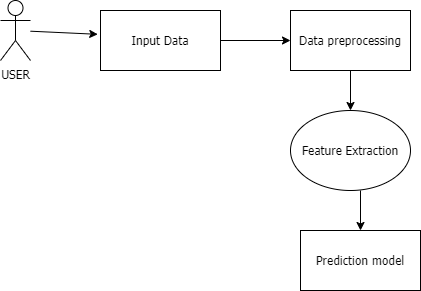
This is basically a contextual diagram, also referred to as a “context diagram”. It only represents the top level or the 0 Level in the whole process.it gives an abstraction kind of view and shows the whole system as a single process and its relationship to externalities.



**Fig 4.4 – Data Flow Diagram 0**

# DATA FLOW DIAGRAM LEVEL 1

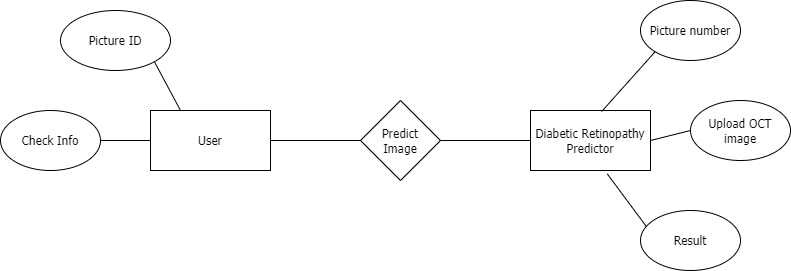
Level DFDs represent the complete system as a single process. it rotates every process and subprocess that comes together in a sequence to form the complete system. This along with) and 2-level data flow diagrams comprise the “fundamental system model”.



**Fig 4.5 – Data Flow Diagram 1**

# 4.5 ER DIAGRAM

ER stands for Entity Relationship. These diagrams display the relationship of entities that are used and stored in the database. They explain the structure of the whole process. These diagrams can be made using three basic concepts, attributes, relationships, and entities.

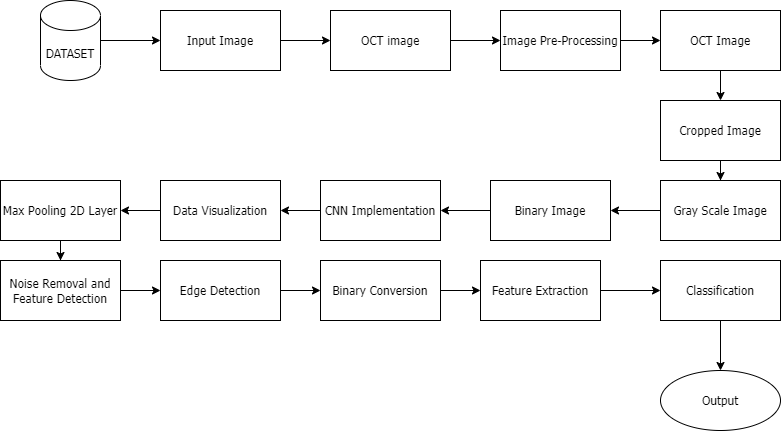


**Fig 4.6 – ER Diagram**

# CHAPTER 5 SYSTEM ARCHITECTURE

## SYSTEM ARCHITECTURE

This diagram is nothing but a simple description of all the entities that have been incorporated into the system. The diagram represents the relations between each of them and involves a sequence of decision-making processes and steps. You can simply call it a visual or the whole process and its implementation. All functional correspondences are explained in this diagram.



## Fig 5.1 – Architecture Diagram

* 1. **MODULE DESCRIPTION**

## MODEL FEATURES

The project that is proposed is a Deep Learning model that is used for identifying the type of infection of due to diabetes or simply excess amount of sugar in human blood. The model has user friendly interface where the user can upload the image of examination and can get the results by user by just pressing 3 buttons. The infected region along with the condition or level of the infection can be identified using this model. The model is well trained by datasets of the infected OCT image which makes the model more accurate and helps producing more perfect result. The model aims to avoid the medical examinations and the

cost involved in identifying the infected eye.

## IMAGE PRE-PROCESSING

Optical coherence tomography (OCT) images are high resolution images, contactless and nondestructive testing property. OCT image are cross sectional image of retina used for classifying diabetic retinopathy more precisely. The main structures that can be visualized on a OCT photo are the central and peripheral retina, optic disc and macula. OCT photography can be performed with colored filters, or with specialized dyes including fluorescein and indocyanine green. Gray scale A grayscale image is simply one in which the only colors are shades of gray. The reason for distinguishing such images from any other sort of colored images are that only little information need is to be given for each pixel. Median filter The median filter is a non-linear digital filtering technique, often used to remove noise from an image or signal. This noise reducing process are typical pre-processing steps in order to improve the results of the later/future processing for example, edge detection present in an image. Morphological Illumination Some morphological transformations are used to detect the unevenly 21 illuminated background of text images characterized by poor lighting, and to acquire illumination normalized result. Lastly, these multi directional images that are pre-processed are merged to produce the final even illumination image. Noise removal and feature detection: Noise removal algorithms are the procedure to possibly remove or reduce the noise that are present in the image which is done basically by smoothening the entire image leaving the regions near contrast levels. But these methods can obscure fine, low contrast details the local appearance around each feature point is described in some way that is (ideally) invariant under changes in illumination, translation, scale, and in-plane rotation. Edge detection: It is an image pre-processing method for identifying the boundaries/borders of the

objects present within given images. It works by detecting discontinuities in brightness. Edge detection is a process of image pre-processing that is used for image segmenting and data extraction in regions such as image processing, cv, and machine vision. Feature extraction: It is a core part that is involved in dimensionality reduction processes, where, an initial set of the raw input/training data is divided and reduced into more manageable group. The important characteristic of these large data sets are they literally have a large number of variables present as information in images which could be difficult for computer for predictions. Binary conversion: Process of converting images to binary form so that the image consists only pixels that can have one of exactly two colors, black and white. Binary images are also commonly referred to as bi-level or two-level.

## NETWORK SELECTION

There are eleven deep learning frameworks which could possibly be used for identifying diabetic retinopathy from OCT images: AlexNet, GoogLeNet , ResNet-18, ResNet-50, VGGNet-16, VGGNet-19, ResNet-101, Inception-v3, Inception-ResNet-v2, DenseNet-201, and SqueezeNet. GoogLeNet, ResNet, DenseNet-201, Inception-v3, Inception-ResNet-v2, and SqueezeNet have complex directed acyclic graph network architectures wherelayers have inputs/outputs to/from multiple layers. AlexNet and VGGNet contain simple series of network architectures. All networks have one input and one output layer. They were originally trained on more than a million natural images from1000 categories in the ImageNet database. we have used VGG16 network for training our model which typically has 16 layers of which 13 are convolutional layers with same padding and stride 1 and 3 dense layers with stride 2. Typically, there are 21 layers in vgg16 network out of which only 16 are weight layers (i.e., Learnable parameters)

## ALGORITHM

VGG16(Visual geometry group) is a convolutional neural network architecture which was used to win ILSVR(ImageNet) competition in 2014.It is considered to be one of the excellent vision model architectures till date. VGG16 architecture has 16 layers. One demerit is that VGG16 trained ImageNet weights is 528mb.So it takes quite a lot of disk space and bandwidth that makes it inefficient in some instances. Relu(Rectified Linear unit) is very simple activation function to calculate, as it involves only a comparison between its input and value 0. It has a derivative of either 0 or 1, depending on whether its input is respectively negative or not. The dropout layers is a mask that nullifies the contribution of some neurons towards the next layer and leaves unmodified all others. Dropout layers are important in training CNNs because they prevent overfitting since they mask some features for every batch of training. If they weren’t present in the model, the first set of train samples greatly impacts the learning by model in a negative/disproportionately high manner. Categorical cross-entropy loss function is a loss function that is used in multi-class classification tasks. These tasks where an example can be only belonged to one out of many possible categories and model must decide which one it is.

# CHAPTER 6

**SYSTEM IMPLEMENTATIONS**

## INTRODUCTION

In this chapter, we will be discussing the actual designing and working of our proposed project. Implementation is the process that must follow any preliminary thinking for some things to occur.

## DATA COLLECTION

Appropriate datasets are required at all stages of object recognition research, starting from the training phase to evaluating the performance of recognition algorithms. There are different steps of collecting data:

* + 1. Scraping from The Web:

Finding and retrieving photos and videos from the internet on a manual basis.

* + 1. Third-party:

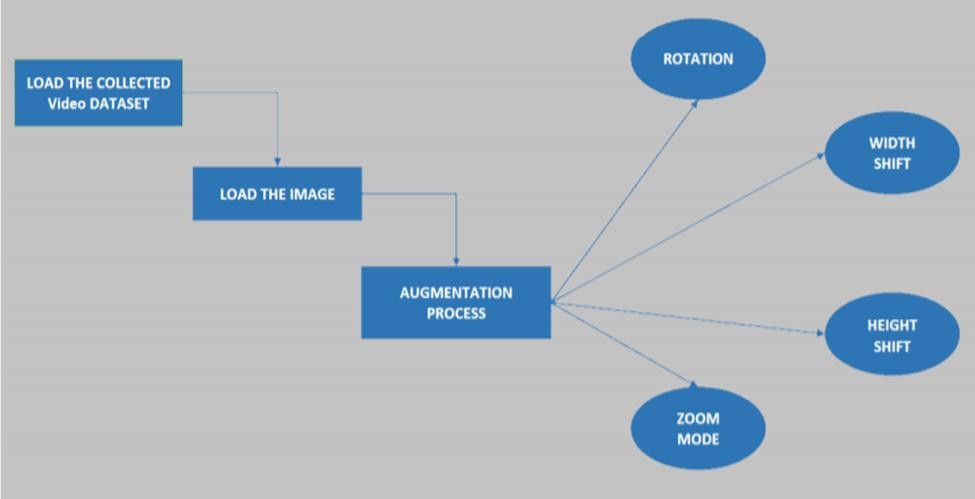
Many start-ups have begun to offer their own image annotation services, as data has become such an asset in the deep learning era. Third- party datasets are what they're called. For our project, all the data are present in the module itself. Hence, we can fetch those trained data to formulate them. We have downloaded dataset of infected OCT images of retina from Kaggle website.

## DATA PREPARATION

Once the data is collected it goes for data preparation where the following process are followed:

## Data Augmentation

Data augmentation is used to maximize the size of the usable dataset. So, all we must do now is make a few small changes to our existing records. Flips, translations, and rotations are examples of slight changes. This process is called Data Augmentation



## Fig 6.1 DATA AUGUMENTATION

* + 1. **Data Cleansing**

Usually, the data present will be of different formats, inaccurate and duplicate data. So, to make all the data to be of same format data cleansing is used.

## TRAINING MODEL

Supervised training: Its use of labelled dataset to train algorithm that to classify data or predict outcomes accurately .Training the model plays a vital role in achieving optimum performance for specific task .Here we have used Adam v2 as training optimizer with initial learning rate of 0.0001 and epochs set to 10 counts .As the initial training option we used batch size of 64.Early stopping is set to 3 to avoid overfitting of training and validation data at each iteration .As the pre-trained models used here were originally trained with natural images that had different image properties ,we resized our images and normalized the color channel(s) as required by the models .We have also used random horizontal and vertical translations of [20,20]pixels and rescaled with 0.1/255 on input image as data augmentation.

## VGG16 - FEATURE EXTRACTOR

Training (ML) 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. ML Or DL algorithms will build a model or architecture based on the sample data, referred commonly as "training data set", in order to make the model do the predictions or some decisions without being externally programmed or inputted by users 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. Neural Network (NN) Neural network is a computational learning system that uses a network of 24 functions to understand and translate a data input of one form into a desired output, usually in another form. The idea of the Artificial Neural Network (ANN) is basically motived by mankind biology and the way the neurons present in the human’s brain work together in order to understand the inputs from human senses. Neural networks are simply one among the many tools and the ways used in ML/DL algorithms. This neural network might itself could be used as piece in many varieties ML/DL algorithms to process critical input data into the area which the computers could understand. Deep Neural Network (DNN) Deep neural network represents the type of machine learning when the system uses many layers of nodes to derive high-level functions from input information. It means transforming the data into a more creative and abstract component. A DNN is a type of Artificial Neural Network (ANN) consists of multitude of layers present in between the input and output layers of a system. Each mathematical workings as such is considered to be a layer of model, and critical or complex DNN contains as many layers as possible, hence called with word the "deep" networks. DNNs can model complex non-linear relationships. Convolution Neural Network (CNN) Convolutional neural networks (CNN) are

mostly used for image recognition, and rarely for audio recognition. These models are more probably applied to the image datasets since it does not require to validate or check all the pixels one by one.

An input OCT image will be provided to model. Later the image will be

pre-processed and necessary features will be extracted by VGG16 model. Convolutional Neural Network (CNN) model validates or checks an input/training image block by block, that starts from the left side upper corner of that image and moves further along pixel by pixel of image until successful completion of check of image pixels. Later the result of all verified pixels are passed along the convolutional layers, in which the data elements only do have connections while others don’t have connections and left out. On the basis of this available data from model, this system will produce the result of the completed verifications and can decides what is present in the picture.

## CLIENT-SIDE CODING App.py

from future import division, print\_function # coding=utf-8

import sys import os import glob import re

from PIL import Image import numpy as np import json

import tensorflow as tf

from tensorflow.compat.v1 import InteractiveSession

from tensorflow.keras.layers import Activation, Flatten, Dropout, Dense from tensorflow.keras.models import Model, Sequential

from tensorflow.keras.applications.vgg16 import VGG16 from tensorflow.keras.optimizers import Adam as adam\_v2

from tensorflow.keras.applications.vgg16 import VGG16 #from tensorflow.keras.optimizers import adam

#config = ConfigProto() #config.gpu\_options.per\_process\_gpu\_memory\_fraction = 0.2 #config.gpu\_options.allow\_growth = True

#session = InteractiveSession(config=config) # Keras

from tensorflow.keras.models import load\_model from tensorflow.keras.preprocessing import image

# Flask utils

from flask import Flask, redirect, url\_for, request, render\_template from werkzeug.utils import secure\_filename

#from gevent.pywsgi import WSGIServer

# Define a flask app app = Flask( name )

# Model saved with Keras model.save() MODEL\_PATH ='retina.h5'

# Load your trained model

model = load\_model(MODEL\_PATH)

def model\_predict(img\_path, model): print(img\_path)

img = image.load\_img(img\_path, target\_size=(224, 224))

# Preprocessing the image

x = image.img\_to\_array(img) # x = np.true\_divide(x, 255) ## Scaling

x=x/255

x = np.expand\_dims(x, axis=0)

# Be careful how your trained model deals with the input # otherwise, it won't make correct prediction!

# x = preprocess\_input(x)

preds = model.predict(x) preds=np.argmax(preds, axis=1) if preds==0:

preds="Choroidal neovascularization" elif preds==1:

preds="Diabetic Macular Edema" elif preds==2:

preds="DRUSEN" else:

preds="Normal"

return preds

@app.route('/', methods=['GET']) def index():

# Main page

return render\_template('index.html')

@app.route('/predict', methods=['GET', 'POST']) def upload():

if request.method == 'POST':

# Get the file from post request f = request.files['file']

# Save the file to ./uploads

basepath = os.path.dirname( file ) file\_path = os.path.join(

basepath, 'uploads', secure\_filename(f.filename)) f.save(file\_path)

# Make prediction

preds = model\_predict(file\_path, model) result=preds

return result

return None

if \_\_name == ' main ': app.run(port=5001,debug=True)

## SERVER-SIDE CODING Retina\_disease.ipynb

!pip install tensorflow==2.2.0

#download dataset

!pip install opendatasets import matplotlib.pyplot as plt import opendatasets as od

od.download("https://[www.kaggle.com/datasets/paultimothymooney/kermany2](http://www.kaggle.com/datasets/paultimothymooney/kermany2) 018")

#visualizing the data import numpy as np import pandas as pd import os

import matplotlib.pyplot as plt print(os.listdir("/content/kermany2018/OCT2017 /test/")) test\_path = "/content/kermany2018/OCT2017 /test/" classes = os.listdir(test\_path)

print('classes: ', classes)

def display\_images(random\_number): for i in classes:

new\_path = test\_path + i

image\_path = new\_path + '/' + os.listdir(new\_path)[random\_number] plt.title(i)

plt.imshow(plt.imread(image\_path)) plt.show()

train\_path = "/content/kermany2018/OCT2017 /train/" classes = os.listdir(train\_path)

print('classes: ', classes)

def display\_train\_images(random\_number): for i in classes:

new\_path = train\_path + i

image\_path = new\_path + '/' + os.listdir(new\_path)[random\_number] plt.title(i)

plt.imshow(plt.imread(image\_path)) plt.show()

val\_path = "/content/kermany2018/OCT2017 /val/" classes = os.listdir(val\_path)

print('classes: ', classes)

def display\_val\_images(random\_number): for i in classes:

new\_path = val\_path + i

image\_path = new\_path + '/' + os.listdir(new\_path)[random\_number] plt.title(i)

plt.imshow(plt.imread(image\_path)) plt.show()

display\_train\_images(500) display\_images(100) display\_val\_images(1) #counting classes and images

def count\_images(path, classes): class\_count = []

for i in classes: class\_count.append(len(os.listdir(train\_path + i)))

df = pd.DataFrame(columns = ["Class\_Name", "No of Images"]) df['Class\_Name'] = classes

df["No of Images"] = class\_count return df

count\_images(test\_path , classes) def count\_images(path, classes):

class\_count = [] for i in classes:

class\_count.append(len(os.listdir(train\_path + i)))

df = pd.DataFrame(columns = ["Class\_Name", "No of Images"]) df['Class\_Name'] = classes

df["No of Images"] = class\_count return df

def count\_images(path, classes): class\_count = []

for i in classes: class\_count.append(len(os.listdir(val\_path + i)))

df = pd.DataFrame(columns = ["Class\_Name", "No of Images"]) df['Class\_Name'] = classes

df["No of Images"] = class\_count return df

count\_images(val\_path , classes)

#image extensions

def image\_extensions(classes):

for label in classes: extensions = []

for img in os.listdir(test\_path + label): extensions.append(img.split('.')[-1])

print("class: ", label)

print("extensions: ", list(set(extensions))) image\_extensions(classes)

#height and width of images def image\_size(classes):

for label in classes: ht, wd, = [], [],

for img in os.listdir(test\_path + label):

im = plt.imread(test\_path + label + '/' + img) ht.append(im.shape[0])

wd.append(im.shape[1])

print("class: ", label)

print("average height is {}, average width is {}:".format(sum(ht)/len(ht), sum(wd)/len(wd)))

image\_size(classes)

#importing tensorflow dependencies

from tensorflow.keras.layers import Activation, Flatten, Dropout, Dense from tensorflow.keras.models import Model, Sequential

from tensorflow.keras.applications.vgg16 import VGG16 from tensorflow.keras.optimizers import Adam as adam\_v2 initial\_lr = 0.0001

epochs = 10

#vgg model structure def vgg\_model():

vgg = VGG16(include\_top = False, weights = 'imagenet', input\_shape = (224, 224, 3))

model = Sequential() model.add(vgg) model.add(Flatten())

model.add(Dense(256, activation = "relu")) model.add(Dropout(0.3)) model.add(Dense(4, activation = 'softmax'))

model.compile(loss = 'categorical\_crossentropy', optimizer = adam\_v2(lr = initial\_lr, decay = initial\_lr/epochs),

metrics = ["accuracy"])

return model vgg16\_model = vgg\_model() vgg16\_model.summary()

#data augmentation

from tensorflow.keras.preprocessing.image import img\_to\_array,

ImageDataGenerator

data\_generator = ImageDataGenerator(rotation\_range = 40,

width\_shift\_range = 0.2,

height\_shift\_range = 0.2, rescale = 1./255, shear\_range = 0.2,

zoom\_range = 0.2, horizontal\_flip = True, fill\_mode = 'nearest', validation\_split = 0.25)

train\_dataset = data\_generator.flow\_from\_directory(train\_path,

target\_size = (224,224), class\_mode = "categorical", batch\_size = 64,

subset = "training")

val\_dataset = data\_generator.flow\_from\_directory(val\_path,

target\_size = (224,224), class\_mode = "categorical",

batch\_size = 64, subset = "validation")

def visualize\_generator\_data(img\_count):

for val in range(img\_count): img, label = val\_dataset.next()

print("image shape is: ", img.shape) plt.imshow(img[0])

print(img[4].shape) plt.show()

visualize\_generator\_data(4)

#training model

from tensorflow.keras.callbacks import ModelCheckpoint, EarlyStopping checkpoint= ModelCheckpoint( 'vggmodel.h5', verbose=1, save\_best\_only=True)

earlystopping = EarlyStopping(patience=3) history = vgg16\_model.fit(train\_dataset,

epochs= epochs, validation\_data = val\_dataset,

callbacks = [checkpoint, earlystopping], verbose = True)

#plotting accuracy plt.plot(history.history["accuracy"]) plt.plot(history.history["val\_accuracy"])

plt.xlabel("Epochs") plt.ylabel("Accuracy") plt.title("VGG 16 Model Accuracy") plt.legend(["Train", "Validation"]) plt.show()

#plotting loss plt.plot(history.history["loss"]) plt.plot(history.history["val\_loss"]) plt.xlabel("Epochs") plt.ylabel("Loss")

plt.title("VGG 16 Model Loss") plt.legend(["Train", "Validation"]) plt.show()

#predictions

predictions = vgg16\_model.predict\_generator(val\_dataset) predictions

# CHAPTER 7 TESTING

## Testing Methodologies

There are several methods or strategies of testing of a project modules as unique components or as a whole or as a byproduct as a part of software testing methodologies. Some of them are discussed below:

## SYSTEM TESTING

Testing is performed to identify errors. It is used for quality assurance. System Test is an core part of an projects development and maintenance lifecycle process. The goal of the testing during this phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself. For example, the design must not have any logic faults in the design before coding commences, otherwise, the cost of fixing the faults will be considerably higher as reflected. Detection of design faults can be achieved by means of inspection as well as a walkthrough.

Testing of project or work is an important part in software development life cycle which could not be ignored for better output. Testing is a process that checks for malfunction or so-called errors, as a whole of the code testing that could possibly involve the following test cases:

* Static analysis is used to investigate the structural properties of the Source code.
* Dynamic testing is used to investigate the behavior of the source code by executing the program on the test data.

## Unit Testing

Unit testing are conducted in order to verify the performance of each unique modular components of the project code. It focuses precisely on the smallest unit of the software design (i.e.), the modules. The white-box test strategies are more frequently heavily deployed for unit testing.

## Functional Tests

These type of test cases are more probably involved in working our code with some nominal input values for which the expected result of that code, as well as the boundary and special values that is logically related inputs and files of identical elements, and empty files are well known by developer or tester.

Three types of tests in Functional test:

* Performance Test
* Stress Test
* Structure Test

## Performance Test

This type of testing particularly involves in the determination of the amount of time spent in executing the various parts of the unit, program outputs, and response time and the utilization level of device by the program unit.

## Stress Test

Stress Test is a test designed to intentionally break the unit. A large amount could possibly be learned about the strengths and the restrictions of a code by probably the examination in the manner in which a programmer in which a program unit breaks.

## Structure Test

This type of testing methodology is more concerned with working of the logic i.e., internal logic of a program and tracing through working of particular execution paths. The possible ways in which the White-Box test strategies are employed to make sure that these test cases could assure or guarantee that all these independent paths within a single module are worked out at least once during this test method.

* Exercise all logics on their true or false sides.
* Execute all loops at their boundaries and within their operational bounds.
* Exercise all the internal data bodies to ensure their validity.
* Checking attributes for their correctness.
* Handling end of file conditions, I/O errors, buffer problems, and textual errors in the output information

## Integration Testing

Integration testing is a testing methodology which composes of a systemized technique for the construction of the code structure as well as at the same time conducts some possible tests to uncover the errors associated with interfaces. i.e., integration testing is the complete testing of the set of modules that makes up the product. The objective of this type of testing is to take the modules that are not tested and then build a code structure tester that must find out the critical modules in program. These Critical modules which are found as a result of this test should be tested as early as possible in life cycle. One approach is to wait until all the units have passed testing, and then combine them and then test. This approach evolved from the unstructured testing of small programs. Another possible strategy is to construct the by product in increments of test units. A limited set of modules are integrated or combined together as a whole and tested once, to which another module can be added and tested in combination. And so on. The advantages of this approach are that interface dispenses can be easily found and corrected.

The major error that was faced during the project is a linking error. When all such modules of code are integrated the links associated are not properly set with all associated support files. Then we have done check for interconnections. Errors are located to the new module and its intercommunications. The product development can be staged, and modules integrated as they complete unit testing. Test process is said to be completed when the last module is combined in some way and tested.

## TESTING TECHNIQUES / TESTING STRATEGIES

A Testing is process of executing a block of code or entire program with the motive to find the existing errors in code or software. A best test case is the one which has the higher probabilities of finding out an not yet discovered errors in the software. A successful test is one that finds out not yet discovered errors in the code. System testing is a testing methodology which performs the implementation stage of program, that is more motivated at assuring that the software works effectively, efficiently and more accurately as the developer expected before the commencements of the live operation. It ensures whether the entire set of programs works as a whole or not. System testing requires a test consisting of several key activities and steps for running a program, string, and system and is important in adopting a successful new system. The last and final possible chance in order to find out and valuate the existing errors before the software is deployed is that for user acceptance testing.

The software testing procedures begins as soon as the software or code is created and the documents and other related data are structured out. Software testing is essential for correcting error. If not done or completed, the software or the project is not said to be in the state of completion and said to be incomplete. Software testing is the crucial part of the so-called process the software quality assurance (SQA) and represents the final review of design specification and code. Testing is the process of running the software with the motive of detecting the

Underlying or covered error. A successful test is one that finds out not yet discovered errors in the code. Any product or software could be tested in one among the two ways described below:

## White-box testing

This type of testing is also referred to as Glass box testing. In this testing, by knowing the specific functions that a product has been designed to

perform tests can be conducted that demonstrate each function is fully operational and at the same time search for errors in each function. It is the test case designing way that uses the control structure of the procedural design to design these test cases for program. Basis path testing is a type of testing that comes under the category of white box testing.

Basis path testing:

* Flow graph notation
* Cyclomatic complexity
* Deriving test cases
* Graph matrices Control

## BASIS PATH TESTING

The established technique of flow graph with Cyclamate complexity was used to derive test cases for all the functions. Steps involved in this type of test are mentioned below:

In order to compute the cyclomatic complexity of the program or software or component we should be using the below mentioned formula:

V (G) =E-N+2 or V (G) =P+1 or

V (G) =Number of Regions

Where V (G) is Cyclomatic complexity, E is the number of edges,

N is the total number of flow graph nodes, P is the number of predicate nodes.

Find out the basis of the set of linearly independent paths.

## Black box testing

Black box test is a test strategy that is done with knowledge of the internal working of a software or code, more probably a test could be performed to

assure that “all gears work out properly as they were intended to do so”, that is the internal working performs according to specs and all the existing internal components are adequately worked out properly. It basically focuses on the functional requirements of the program.

Steps in black-box test case designing are:

* Graph-based testing methods
* Equivalence partitioning
* Boundary value analysis
* Comparison testing

## SOFTWARE TESTING STRATEGIES:

The software testing strategy gives us a so-called road map for software developer. Testing as a whole is a combination of certain activities carried out that can be planned earlier in advance and those activities are conducted systematically. For these particular reasons, a framework for the software testing is some set of steps into which we could place some specific test cases that must strategy must have the following characteristics:

* Testing begins at the module level and works “outward” toward the integration of the entire computer-based system.
* Different testing techniques are appropriate at different points in time.
* The developer of the software and an independent test group conducts testing.
* Testing and Debugging are different activities but debugging must be accommodated in any testing strategy.

## Integration Testing

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with it. Separate modules or code blocks, which are highly vulnerable to interfacing error, shouldn’t be guessed to function quickly as we put them

together. The problem, of course, is “putting them together”- interfacing. There may be the chances of data loss across another’s sub-functions when combined may not produce the desired major function; individually acceptable impressions may be magnified to unacceptable levels; global data structures can present problems.

## Program Testing

The logical and syntax errors have been pointed out by program testing. A syntax error is an error in a program statement that violates one or more rules of the language in which it is written. An improperly defined field dimension or omitted keywords are common syntax errors. Such errors that are displayed only by means of error messages generated by the system. A logic error on the other hand deals with the incorrect data fields, out-of-range items, and invalid combinations. Since the compiler will not deduct logical errors, the programmer must examine the output. Condition testing is a testing method that works out the logics presented in a particular module or block of code. The possible types of elements in a condition include a Boolean operator, Boolean variable, a pair of Boolean parentheses A relational operator or an arithmetic expression. This condition test focuses majorly on the testing of each and every single condition or logic present in the entire code. The purpose of the condition test is to deduct not only errors in the condition of a program but also other errors in the program.

## Security Testing

Security testing attempts to verify the protection mechanisms built into a system well, in fact, protect it from improper penetration. The system security must be tested for invulnerability from frontal attacks must also be tested for invulnerability from rear attacks. During security, the tester places the role of the individual who desires to penetrate the system.

## Validation Testing

While the proceeding of the validation testing, the program or final code is entirely arranged or assembled as a whole as a package. Interface errors are more probably been fixed and uncovered and corrected and a final set of software testing the validation test begins after that. Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in a manner that is reasonably expected by the customer. Software validation is achieved through a series of black-box tests that demonstrate conformity with requirements. After the validation test has been conducted, one of two conditions exists.

* The function or performance characteristics conform to specifications and are accepted.
* A validation from the specification is uncovered and a deficiency is created.

Deviations or errors discovered at this step in this project are corrected prior to completion of the project with the help of the user by negotiating to establish a method for resolving deficiencies. Thus, our proposed system under some of these considerations has been successfully tested by the use of validation test and is found to be working perfectly overall. Though there existed some defects or deficiencies in proposed system which were not drastic and could be solved later or ignored too.

## User Acceptance Testing

User acceptance is a type of testing of the proposed system and is a vital factor for the success of any system not only our work. The system under consideration is tested for user acceptance by constantly keeping in touch with

prospective systems and users at the time of developing and making changes whenever required. This is done in regard to the following points.

* Input screen design.
* Output screen design.

# TEST CASES& REPORT:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.NO** | **ACTION** | **INPUTS** | **EXPECTED**  **OUTPUT** | **ACTUAL**  **OUTPUT** | **TEST**  **RESULT** |
| 1. | NORMAL DR  DETECTION | Input any normal DR  OCT image | Prediction to be NORMAL DR | Predicted NORMAL  DR | Pass |
| 2. | DRUSEN DR DETECTION | Input any DRUSEN DR OCT  image | Prediction to be DRUSEN DR | Predicted DRUSEN DR | Pass |
| 3. | DME DR  DETECTION | Input any DME DR  OCT image | Prediction to be DME DR | Predicted DME DR | Pass |
| 4. | CNV DR DETECTION | Input any CNV DR  OCT image | Prediction to be CNV DR | Predicted CNV DR | Pass |
| 5. | FAILED TO INPUT ANY IMAGE AND SELECT  PREDICT | No Image | No change | NO change happened | Pass |
| 6. | INPUT ANY IMAGE OTHER  THAN DR | Any images | Error or invalid image display | Predicts some DR type | Fail |

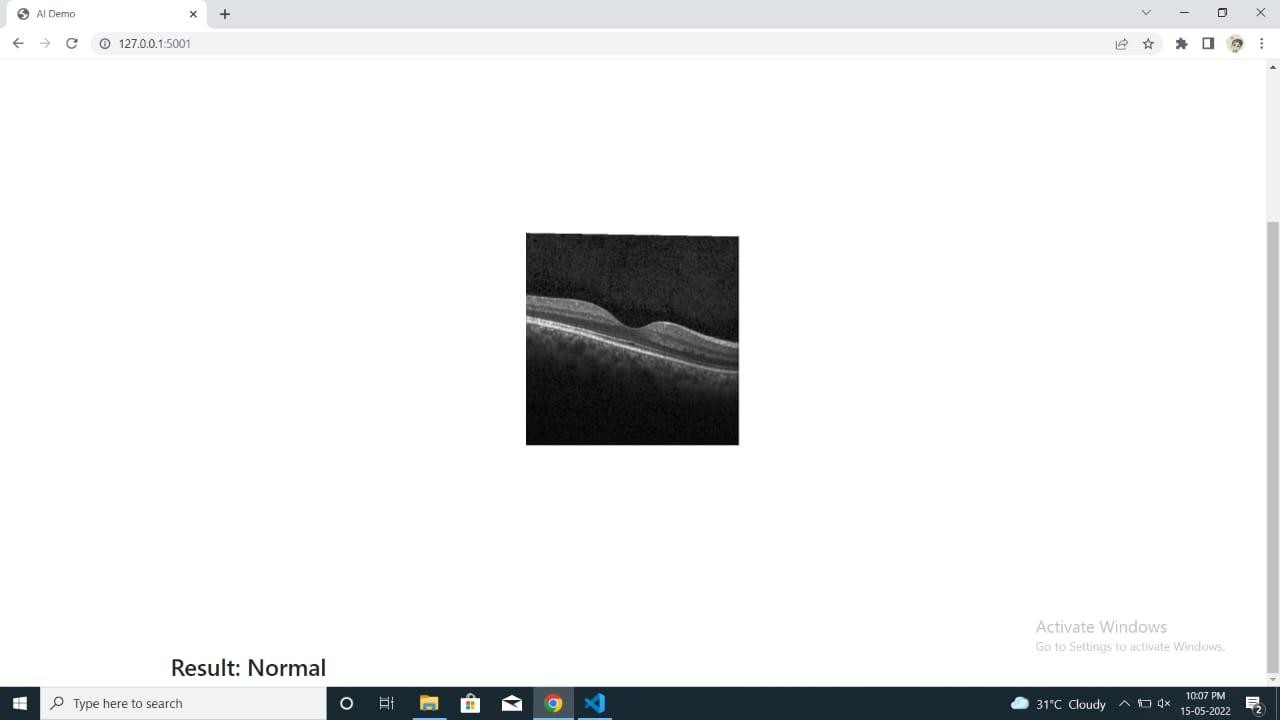
**Tab 7.1 Test case**

# CHAPTER 8

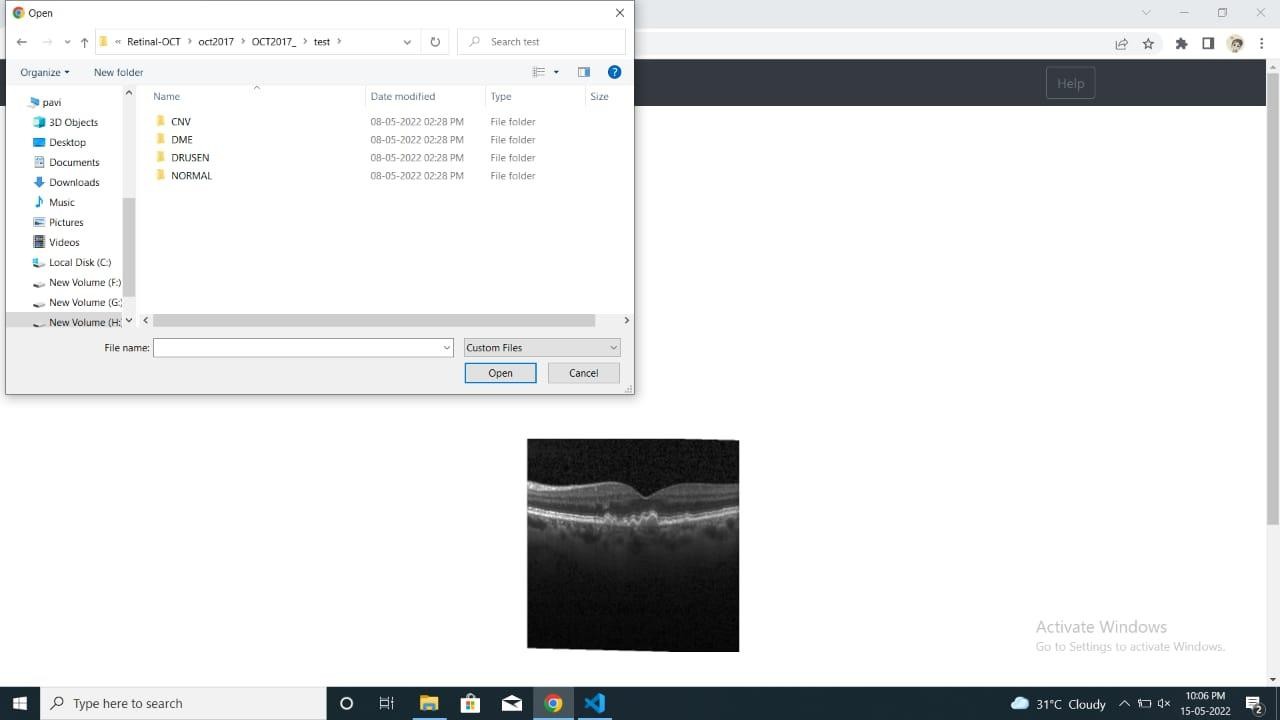
**CONCLUSION AND FUTURE ENHANCEMENTS**

Thus, from our proposed system, we are able to eliminate many complexities faced by conventional detection systems. Our proposed system more possibly reduces the possibility of delays and any other inefficiencies that remained in existing system. Automating these screening processes markably reduces the time and effort required to predict the type of disease, saving some effort and finance for ophthalmologists and by default concludes in the timely treatment of patients. Automation of these systems for the detection of DR plays a vital role in possibly finding out Diabetic retinopathy at an earlier possible or treatable stage. The DR types are based on the type of bulging, blood leakage or hemorrhage that appear in the retina. From our report we have reviewed the best possible automated system for detection and classification of diabetic retinopathy in which it uses deep learning techniques. The common OCT DR datasets that are publicly available have been described, and deep-learning techniques have been briefly explained. Most of the current world researchers use the Convolutional neural network (CNN) for the process of detection and classification of the DR using OCT images due to its efficiency. This review has also discussed the best possible technique that could possibly be utilized to detect and classify DR using CNN. The project is about proposing an optimal model for Diabetic Retinopathy detection. Processing of Retinopathy images is very essential to get proper features. Statistical values can predict level of severity properly but in case of noisy images the chances of getting poor data will lead to lower accuracy. For yielding accurate result, selecting proper features out of the image is also important.

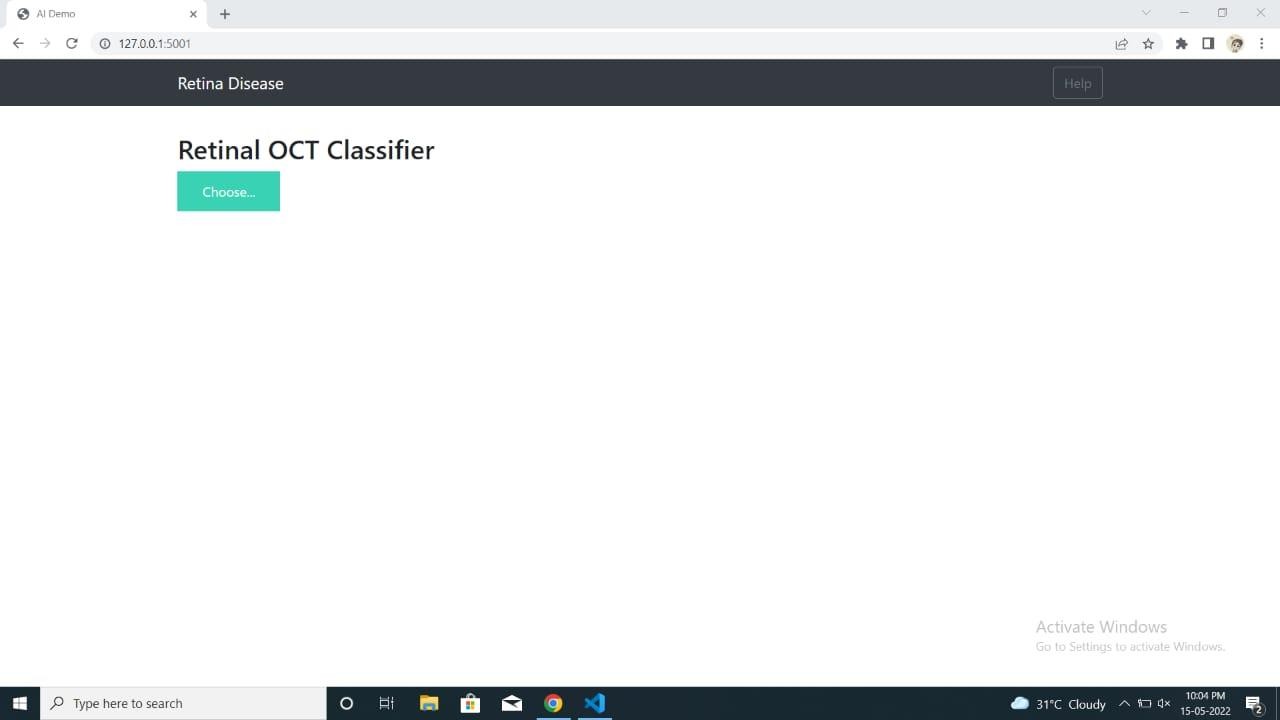
# APPENDIX 1



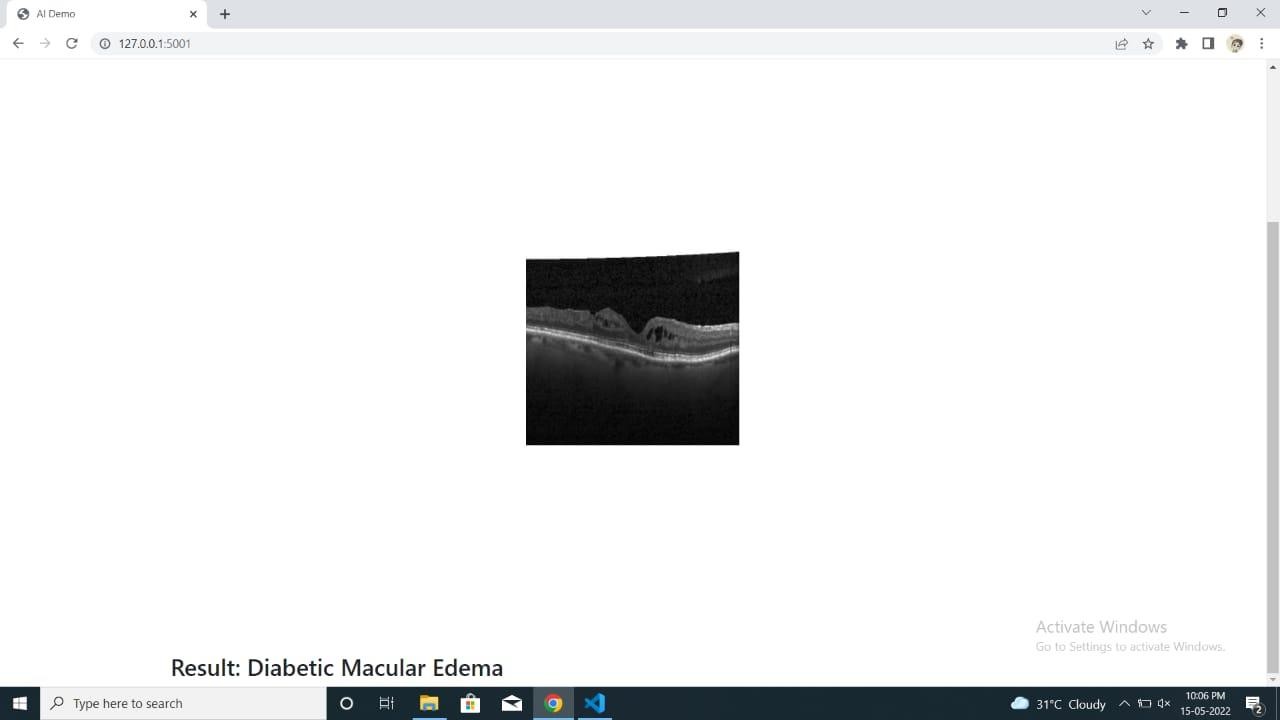
**Fig 9.1 NORMAL IMAGE DETECTED**



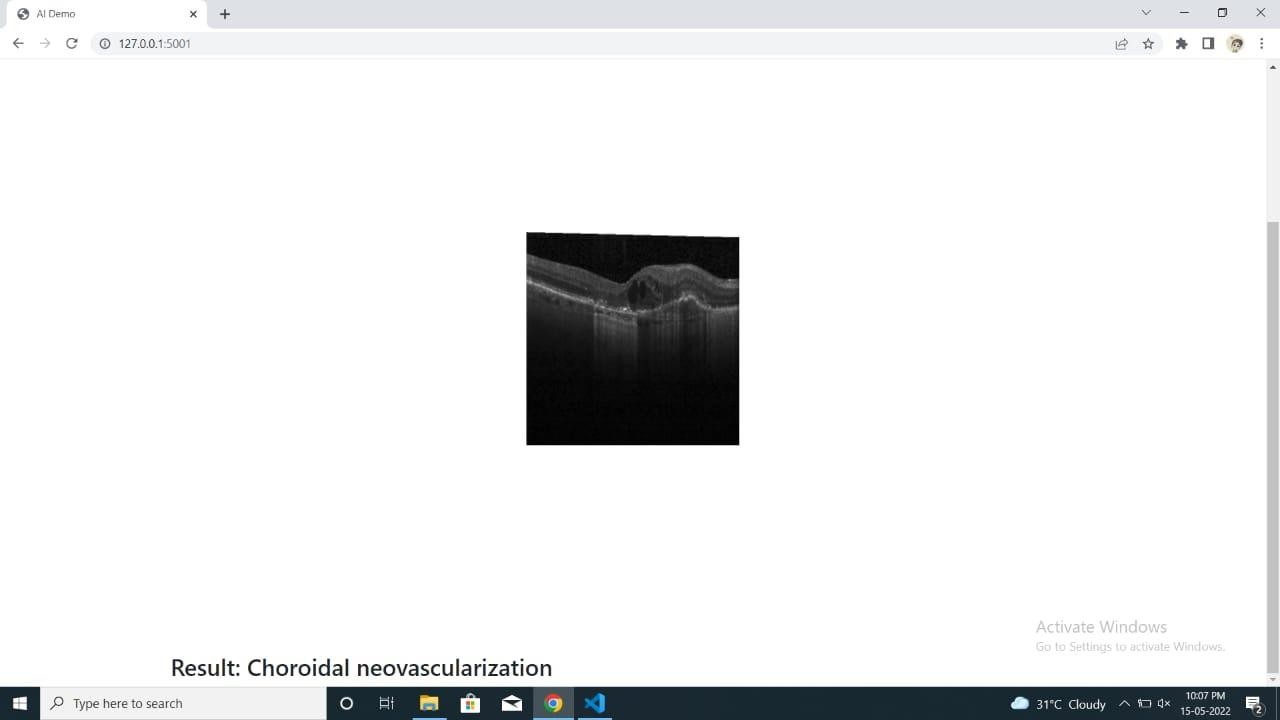
**Fig 9.2 IMAGE UPLOADING UI**



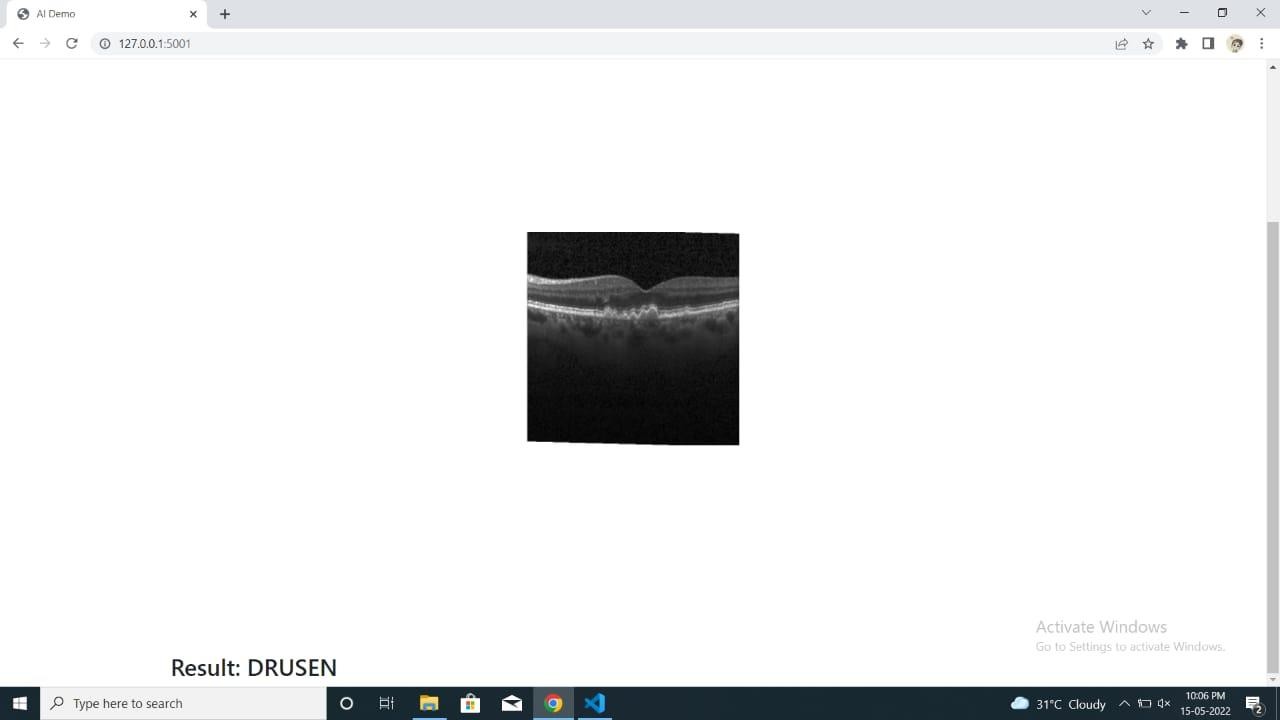
**Fig 9.3 HOME PAGE**



**Fig 9.4 DIABETIC MACULAR EDEMA DETECTED**



**Fig 9.5 CHOROIDAL NEOVASCULARIZATION DETECTED**



**Fig 9.6 DRUSEN DETECTED**

# REFERENCES

1. Aryan Kokane, Gourhari Sharma, Akash Raina, Shubham Narole, Prof. Pramila M. Chawan, "Detection of Diabetic Retinopathy using Machine Learning" in International Research Journal of Engineering and Technology (IRJET), 2020.
2. Wejdan L.Alyoubi, Wafaa M.Shalash, Maysoon F.Abulkhair, "Diabetic retinopathy detection through deep learning techniques: A review" in Informatics in Medicine Unlocked, 2020.
3. Yung-Hui Li, Nai-Ning Yeh, Shih-Jen Chen and Yu-Chien Chung, "Computer- Assisted Diagnosis for Diabetic Retinopathy Based on Fundus Images Using Deep Convolutional Neural Network" in Mobile information systems, 2019.
4. Suvajit Dutta, Bonthala CS Manideep, Syed Muzamil Basha, Ronnie D. Caytiles and N. Ch. S. N. Iyengar, "Classification of Diabetic Retinopathy Images by Using Deep Learning Models" in International Journal of Grid and Distributed Computing, 2018.
5. T Chandrakumar, R Kathirvel, "Classifying Diabetic Retinopathy using Deep Learning Architecture" in International Journal of Engineering Research & Technology (IJERT), 2016.
6. J.Lachure, A.V.Deorankar, S.Lachure, S.Gupta, R.Jadhav,

―Diabetic Retinopathy using Morphological operations and Machine Learning‖, IEEE International Advance Computing Conference (IACC), (2015).

1. R.Priya, P.Aruna, ―SVM and Neural Network based Diagnosis of Diabetic Retinopathy‖, International Journal of computer Application
2. S.Giraddi, J Pujari, S.Seeri, ―Identifying Abnormalities in the Retinal Images using SVM Classifiers‖, International Journal of Computer Applications (0975- 8887), Volume 111 – No.6, (2015).
3. G.Lim, M.L.Lee, W.hsu, ―Transformed Representations for Convolutional Neural Networks in Diabetic Retinopathy Screening‖, Modern Artificial Intelligence for Health Analytic Papers from the AAAI (2014).
4. Vesna Zeljkovi et al, ―Classification Algorithm of Retina Images of Diabetic patients Based on Exudates Detection‖, 978-1-4673-2362-8/12, IEEE (2012)