CHAPTER 1

INTRODUCTION

Artificial intelligence (AI) has proved to be applicable in multifarious fields, including medical tests and diagnostics. For instance, in microscopic examinations, AI can reliably predict certain fluorescent labels on transmitted light microscopy images of unlabeled, fixed, or live biological samples[1]. In ophthalmology, AI can correctly identify diseases as accurately as specialists[2]. Medical imaging provides vital clues for diagnosing doctors and with the ageing population.

Retinal diseases are known causes of visual impairment and blindness and they have been the subject of research for many organizations. In most parts of the developing world have emphasized on avoidable causes of blindness which are more common. Efforts have been made for early treatments for these diseases. There is a need for a system which can provide easier, more cost-effective and achieve higher accuracy with advance technology available in the current time. Health care managers have invested more in preventing or treating blindness from diseases that are easily handled than procuring advance technology towards the treatment of diseases that were considered less common and more expensive to handle and the prevalence of age-related retinal degeneration diseases which are the leading causes of blindness is further increasing[3]. Optical coherence tomography, as a noninvasive imaging modality, plays a major part in guiding the administration of these diseases. Fully automating image detection can significantly decrease the tedious clinician labour and obtain a faithful pre-diagnosis from the analysis of the structural elements of the retina. Thereby, we explore the deep learning methods to do the task automatically and efficiently.

We introduce a Retinal Disease Detection System capable of classifying images. And a convolutional neural network (CNN) convolves an input image with a defined weight matrix to extract specific image features without losing spatial arrangement information. Our project will use an optical coherence tomography (OCT) images on a deep learning framework for screening patients with choroidal neovascularization (CNV), diabetic macular edema (DME), and drusen.

1.1 RATIONALE

The Optical coherence tomography is a noninvasive imaging technique that has been used increasingly to diagnose and manage a variety of retinal diseases and glaucoma. Optical coherence tomography (OCT) is based on the principle of Michelson interferometer. Interference patterns produced by low coherence light reflected from retinal tissues and a reference mirror are processed into an "A-scan" signal. Multiple A-scan signals are aligned to produce a two-dimensional image that can be thought of as a form of "in vivo histology." Optical coherence tomography has been used to identify any irregularities in the retina. It can be used to identify and quantify macular edema, and to measure retinal thickness changes in response to therapy. Optical coherence tomography is an accurate and reproducible method to measure retinal nerve fibre layer thickness. Particularly, when used in combination with other optic nerve imaging techniques, it can be used to differentiate glaucomatous eyes from normal eyes. Despite the above-mentioned usefulness OCT has its limitations that the analysis of the imaging depends upon the operator testing it. To overcome the limitations we used deep learning technique to train the machine for classifying the OCT image with higher accuracy and to detect the disease at an early stage. From the data compiled over the years and the recent surge in the computations powers, it was possible to train and test the model. With the machine making decisions, human error is reduced and the chances of early detection of the disease have increased manifold. Today deep learning and artificial intelligence have boosted the way medical researches take place, nowadays the time taken to analyze genes has reduced very much. Thus, automation in the process of analyzing the OCT images will be advantageous. Statistics state that the frequency of Drusen in the younger population aged 18-54 is high, same goes for the other retinal diseases, the error percentage in such kind of cases is small but considerable so the automation in the process of detecting these diseases is the motivation of developing the deep learning-based system.

1.2 PROBLEM DEFINITION AND PROPOSED SOLUTION

PROBLEM DEFINITION

Retinal diseases vary widely, but most of them cause visual symptoms. Retinal diseases can affect any part of your retina, a thin layer of tissue on the inside back wall of your eye. The retina contains millions of light-sensitive cells (rods and cones) and other nerve cells that receive and organize visual information. The retina sends this information to the brain through the optic nerve, enabling you to see. Diagnosis of

the diseases of the retina has now become a must because many people are likely to become blind if these diseases are not diagnosed at an early stage. Retinal images can be used to diagnose many retina related eye diseases. Early detection and treatment of the diseases will greatly reduce the chances of people becoming blind.

The 4 categories of retinal diseases can be summarized as:

1. Choroidal neovascularization (CNV): Choroidal neovascularization is the creation of new blood vessels in the choroid layer of the eye. Choroidal neovascularization is a common cause of neovascular degenerative maculopathy commonly exacerbated by extreme myopia, malignant myopic degeneration, or age-related developments.

2. Diabetic Macular Edema (DME): DME is a complication of diabetes caused by fluid accumulation in the macula that can affect the fovea. The macula is the central portion in the retina which is in the back of the eye and where vision is the sharpest. Vision loss from DME can progress over months and make it impossible to focus clearly.

3.Drusen: Drusen are yellow deposits under the retina. Drusen are made up of lipids, a fatty protein. Drusen likely do not cause age-related macular degeneration (AMD). But having drusen increases a person's risk of developing AMD. Drusen are made up of protein and calcium salts and generally appear in both eyes.

4. Normal: Normal vision occurs when light is focused directly on the retina rather than in front of or behind it. A person with normal vision can see objects clearly near and far away. There is a need for a system that can be used to detect the above classes of retinal diseases with good accuracy. There is a requirement of a system that can detect the diseases in their early stages and reduced the cost and effort for the detection process.

PROPOSED SOLUTION

In this work, our approach used an ensemble of four classification model instances. The neural network architecture mainly consisted of convolutional layers, pooling layers, and fully connected layers. The convolutional layers extracted features and transformed input images into hierarchical feature maps. The pooling layers (incorporating max pooling and average pooling) merged semantically similar features into one to reduce the dimensionality of the extracted features. The fully connected layers combined these features and produced an image-level classification.

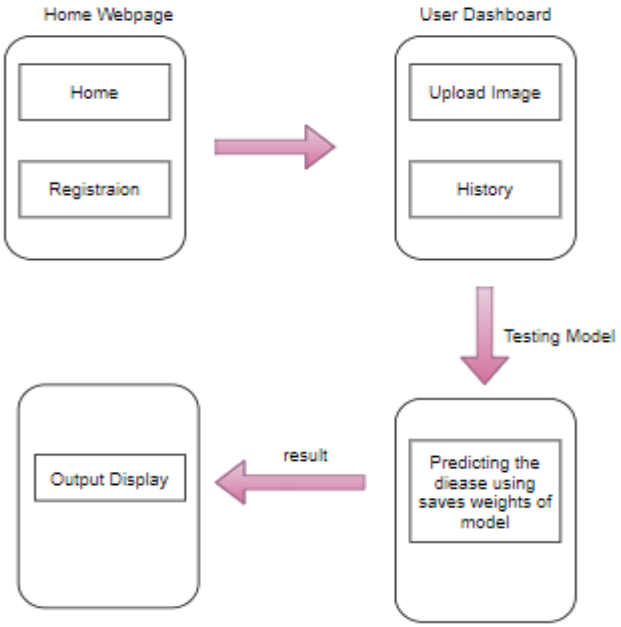


Figure 1.1 The communication mechanism between various modules.

1.3 PROCESS FLOW OF EXISTING SYSTEM

There are many existing systems which has proposed their solution. The existing system uses a traditional approach basis on the doctor's knowledge and his skills to identify the disease.

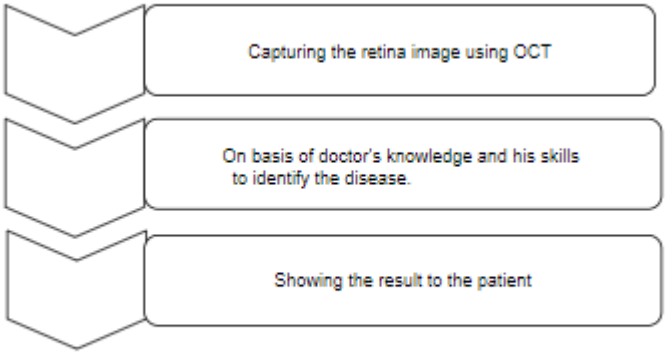


Figure 1.2 Process Flow of Existing System

1.4 LIMITATIONS OF EXISTING SYSTEM

This is one of medical diagnosis of retinal oct traditional apporach.

- This system requires the eye specialist and doctors
- This system is costly and required a very large time to give the result to the patient.
- The chances of proper diagnosis abilities

1.5 OBJECTIVE & SCOPE

Our objective is to create a system that will predict retinal damages as early as much as possible and suggest the best doctor to the user We also giving a personal dashboard patients to save his past upload eyes images.

Our main objective is that, with the help of Machine Learning techniques and the advance technology capable of producing High-resolution images of the retina, it is now possible to detect diseases before they become too hard to cure. Early detection and treatment of the diseases will greatly reduce the chances of permanent damage to the retina of the patients. In this project, we will apply image processing and classification on the retinal OCT images using Convolutional Neural Network and classify different types of retinal disease such as Choroidal neovascularization, Diabetic Macular Edema, Drusen. The early and accurate detection of retinal disease will greatly benefit mankind. This will enable doctors to start the treatment of the detected disease before it became too serious. It will also benefit the patients as the diagnosis of retinal disease in an early stage will reduce the cost of treatment.

1.6 REPORT ORGANISATION

This thesis is organized as follows.

- **Chapter 1** The introduction to the thesis is given in this section and it describes the objectives, motivation and justification.
- **Chapter 2** reviews numerous existing and emerging technologies that are related to the work presented in this thesis under the title literature survey.
- **Chapter 3** Analysis of the whole work is done by explaining the detailed work performed in the project.

- **Chapter 4** System Design of the whole work which includes the UML diagrams such as Use Case, Activity Diagram and Sequence diagram.
- **Chapter 5** describes the various results that are obtained after the complete implementation and screenshots of the project.
- **Chapter 6** includes the testing part of the project.
- **Chapter 7** Includes the functional and non-functional requirements.
- **Chapter 8** Draws conclusions from the work described in previous chapters and discusses possibilities for future development.

Finally, in the last section i.e. Appendix, we have provided the user manual and the references. This section is an exhaustive list of the journals and research papers referred to during the preparation of the thesis.

CHAPTER 2

LITERATURE SURVEY

2.1 BACKGROUND

The earliest approach to detect and classify retinal diseases from images included multiple image processing techniques followed by feature extraction and classification. One such automated technique I ncluded finding abnormalities such as micro-aneurysms, haemorrhages, exudate and cotton wool spot f rom Retinal images This approach uses a noise reduction algorithm and blurring to branch out the four-c lass problem to two cases of a two-class problem. From there on, background subtraction followed by shape estimation to extract important features is used. Finally, those features were used to classify each of the four abnormalities. Similarly, another such feature-based In ophthalmology a technique called Spectral Domain Optical Coherence Tomography is used for viewing the morphology of the retinal layers. Moreover, depth-resolved tissue formation data encoded in the magnitude and delay of the backscatteredd light by spectral analysis is also used to treat these diseases . Though the image is retrieved through this process, the differential diagnosis is conducted by an ophthalmologist. Consequently, there will always be room for human error while performing the differential. Hence, an expert system is required to clearly distinguish between different retinal diseases with fewer mistakes. One of the major reasons for misclassification is due to the stark similarity between Diabetic Retinopathy and other retinal diseases. They can be grouped by three major categories, i) Diabetic Macular Edema (DME) and Age-related degeneration of retinal layers (AMD), ii) Drusen, a condition where lipid or protein build-up occurs in the retinal layer and iii) Choroidal Neovascularization (CNV), a growth of new blood vessels in sub-retinal space. Diabetic Retinopathy and Age-related Macular Degeneration are the most likely cause of retinal diseases worldwide. While Drusen acts as an underlying cause that can trigger DR or AMD in a prolonged timeframe. On the other hand, Choroidal Neovascularization is an advanced stage of age-related macular degeneration that affects about 200,000 people worldwide every year.

2.2 RELATED WORK

2.2.1 Traditional Image Analysis

The earliest approach to detect and classify retinal diseases from images included multiple image processing techniques followed by feature extraction and classification. One such automated technique included finding abnormalities such as micro-aneurysms, haemorrhages, exudate and cotton wool spot from Retinal images This approach uses a noise reduction algorithm and blurring to branch out the four-class problem to two cases of a two-class problem. From there on, background subtraction followed by shape estimation to extract important features is used. Finally, those features were used to classify each of the four abnormalities. Similarly, another such feature-based technique was used for detecting Diabetic Macular Edema and Choroidal Neovascularization. The images were manipulated focused on five distinct parameters: Retinal Thickness, augmentation of Retinal Thickening, Macular volume, retinal morphology and vitreoretinal relationship Other approaches combined statistical classification with edge detection algorithms to detect sharp edges. Sanchez et al.'s algorithm achieved a sensitivity score of 79.6% for classifying Diabetic Retinopathy. Ege et al.'s approach incorporating Mahalanobis classifier detected microaneurysms, haemorrhages, exudates, and cottonwool spots with a sensitivity of 69, 83, 99, and 80%, respectively. It's quite evident that each of these techniques shown slight improvements, but in terms of precision it didn't achieve desired results.

2.1.1 Segmentation Based Approch

The most notable way to identify a patient having Diabetic Macular Edema is the enlargement of macular density in retinal layer. Many approaches have been proposed and implemented that involves segmentation of retinal layers. Further identification of likely causes is also performed for the build-up of liquids in the sub-retinal space. In the authors proposed the idea of segmenting the intraretinal layers in ten parts and then extracted the texture and depth information from each layer. Subsequently, any aberrant retinal features are detected by classifying the dissimilarity between healthy retinas and the diseased ones. Niemeijer et al. introduced a technique for 3D segmentation of regions containing fluid in OCT images using a graph-based implementation. A graph-cut algorithm is applied to get the final predictions from the information initially retrieved from layer-based segmentation of fluid regions. Even though implementation based on a previous segmentation of retinal layers have reported high scoring prediction results, the initial step is reportedly troublesome and erroneous As reported in retinal thickness measurements obtained by different systems has stark dissimilarity. Therefore, it is not quite effective to compare between different retinal depth information retrieved by separate machines. Enforcing the fact that segmentation based approaches weren't as effective as a universal retinal disease recognition system.

CHAPTER 3

SYSTEM REQUIREMENT SPECIFICATION

System analysis patterns or analysis are conceptual models, leading to specifications of a new system. An analysis is a detailed study of various operations performed by a system and their relationships within and outside the system. During analysis, data are collected on the available files, decision points and transactions handled by the present system.

3.1 HARDWARE REQUIREMENTS

Processor	:	Pentium, AMD or Higher Version
Operating System	:	Windows XP/ Windows 7/ Linux
RAM	:	256 MB, 2GB recommended
Hard disk	:	10 GB or more
Display	:	Standard Output Display

3.2 SOFTWARE REQUIREMENTS

Technology Implemented	:	Deep Learning
Language Used	:	Python
Database	:	MongoDB
User Interface Design	:	HTML5,CSS, JavaScript
Web Browser	:	Google Chrome, Internet explorer

CHAPTER 4

SYSTEM ANALYSIS AND DESIGN

Based on the user requirements and the detailed analysis of a system, the system is designed. This is the phase of system designing. It is a crucial phase in the development of a system. It includes the UML diagrams.

4.1 SYSTEM ANALYSIS

System analysis is conducted to study a system or its parts to identify its objectives. It is a problem-solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish its purpose. Here majorly we have developed our whole system into two modules:

The first module is signed in a module that gives personal dashboard experience to the patient.

And the second module will diagnosis the disease on the input image upload by the user.

4.1.1 REQUIREMENT ANALYSIS

Requirement analysis is identifying the needs which are needed to generate the desired output from the system. This requirement can be functional and non-functional.

The functional requirement includes :

- Our Project will give a personal dashboard experience to the patient.
- Our Project will identify the nature of retinal damages and show the output to the patient.

The non-functional requirement includes:

- The system has better speed, faster testing on the given image with low cost and gives the better user interface experience to the patient.

4.1.2 OBJECT ORIENTED ANALYSIS

Object-Oriented Analysis is the first technical activity that is performed as a part of object-oriented software engineering. Object-Oriented Analysis introduces new concepts for examining a problem. It is grounded in a set of basic principles, which are as follows –

- The information domain is modelled.
- Behaviour is represented.
- The function is described.
- Data, functional and behavioural models are divided to uncover greater detail.
- Early models represent the essence of the problem, while later ones provide implementation details.

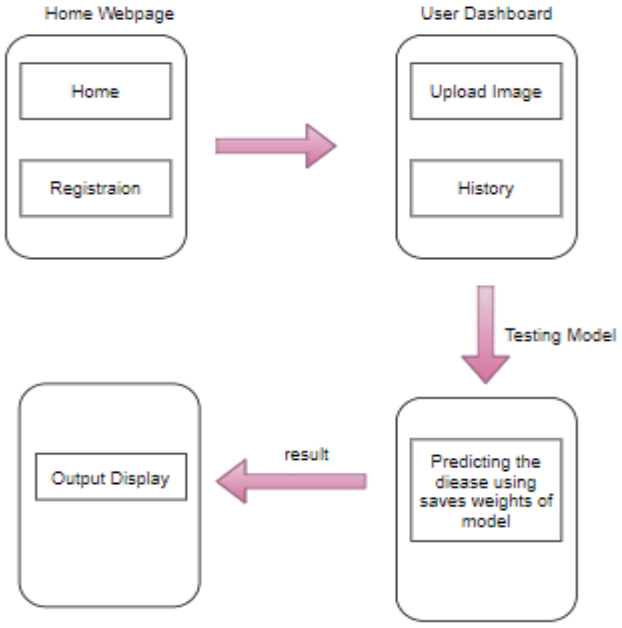


Figure 4.1 The communication mechanism between various modules

4.1.3 ARCHITECTURAL SPECIFICATION

A description of a software and electronics system in terms of its hardware and software components and their interactions. In (hardware, software, or enterprise) systems development, an architectural specification is the set of documentation that describes the structure, behaviour, and more views of that system.

The architectural specification includes the report organization of the project.

The role of the system architectural structure that serves as the blueprint for the target system.

- Allocate required system functionality to hardware and software.
- Document crucial design constraints, assumptions and rationales.

- Support early analysis to make sure that the design approach is highly visible and open to peer review and progressive improvement.
- Demonstrate compliance with system requirements. The SAAS provides an authoritative reference for detailed traceability analysis.
- Explain how the end product will exhibit required qualities such as usability, performance, modifiability, safety and security.
- Describe design management strategies to be used to control the development of the design including the nomination of design patterns and rules.
- Support project planning and budgeting
- Support preparation of acquisition documents (for example, requests for proposal and statements of work)
- Support on-going maintenance and enhancement.

4.1.4 FEASIBILITY ANALYSIS

Feasibility analysis (FA, also called feasibility study) is used to assess the strengths and weaknesses of a proposed project and present directions of activities which will improve a project and achieve desired results. The nature and components of feasibility studies depend primarily on the areas in which analyzed projects are implemented.

The evaluation of legal requirements concerns basically the rules of publishing and using data, software and methods. A part of the rules concerning publishing spatial data (mostly future) can be found in the INSPIRE directive and in copyright laws. Most data and software include information on the form of publishing in metadata or websites where such data is published.

Technical Feasibility

Technical feasibility assesses the current resources (such as hardware and software) and technology, which are required to accomplish user requirements in the software within the allocated time and budget. For this, the software development team ascertains whether the current resources and technology can be upgraded or added in the software to accomplish specified user requirements. Technical feasibility also performs the following tasks.

Analyzes the technical skills and capabilities of the software development team members. Determines whether the relevant technology is stable and establishes. Ascertains that the technology chosen for software development has a large number of users so that they can be consulted when problems arise or improvements are required.

Economic Feasibility

- The cost of data or software procurement,
- The cost of employing a workforce for specific tasks,
- The cost of possible improvement of the qualifications of the workforce employed.

The evaluation of requirements related to the schedule concerns the estimation of the time necessary to complete respective parts of a project, e.g. data procurement.

Feasibility analysis is achieved by checking the load over the application i.e. if 20-25 no. of user login at the same time the system does not crash. The economic requirement includes:

- No hardware used thus hardware cost is reduced.
- The small app does not require any workforce.
- No highly professionals are required. Basic knowledge of java and android is sufficient to build the app.

Operational Feasibility

Operational feasibility assesses the extent to which the required software performs a series of steps to solve business problems and user requirements. This feasibility is dependent on human resources (software development team) and involves visualizing whether the software will operate after it is developed and be operative once it is installed. Operational feasibility also performs the following tasks**.** Determines whether the problems anticipated in user requirements are of high priority. Determines whether the solution suggested by the software development team is acceptable. Analyzes whether users will adapt to new software.

4.1.5 DEVELOPMENT METHOD

After designing the new system, the whole system is required to be converted into computer understanding language. Coding the new system into computer programming language does this. It is an important stage where the defined procedures are transformed into control specifications by the help of a computer language. This is also called the programming phase in which the programmer converts the program specifications into computer instructions, which refer as programs. The programs coordinate the data movements and control the entire process in a system.

Our development occurred in the following phases :

- Platform selection : Python, MongoDB, HTML, CSS , JavaScript
- Observed a tutorial site of

  https://www.pythonforbeginners.com/basics/

  https://www.guru99.com/python-tutorials.htm

  https://www.learnpython.org/
- Created drawings showing the future system.

4.2 SYSTEM DESIGN

4.2.1 USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language is a type of behavioural diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. Use Case diagrams are formally included in two modelling languages defined by the OMG: the Unified Modeling Language (UML) and the Systems.

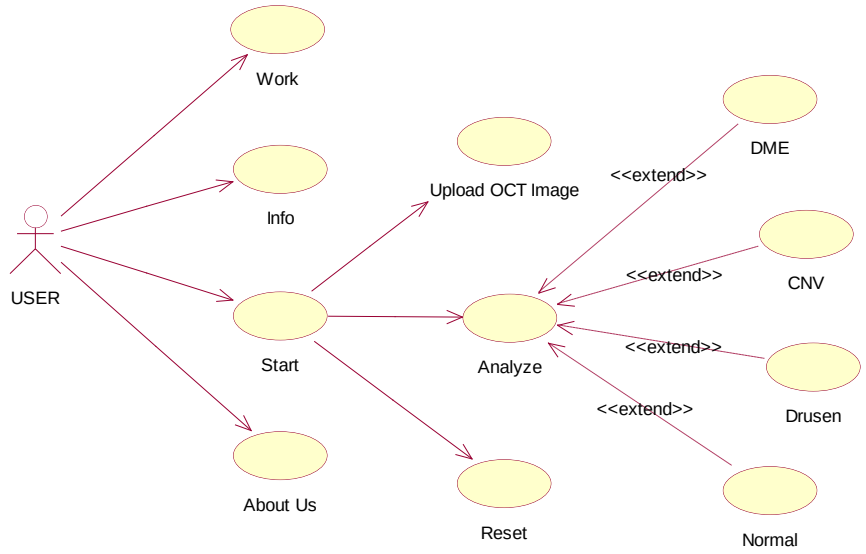


Figure 4.2 Use Case Diagram

4.2.2 ACTIVITY DIAGRAM

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. Activity diagram is basically a flow chart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. Activity diagrams provide a way to model the workflow of a business process. Activity diagrams can also be used to model code-specific information, such as a class operation. Activity diagrams are very similar to a flowchart because of modelling workflow from activity to activity. An activity diagram is basically a special case of a state machine in which most of the states are activities and most of the transitions are implicitly triggered by the completion of the actions in the source activities. Each activity represents the performance of a group of actions in a workflow. Once the activity is complete, the flow of control moves to the next activity or state through a transition.

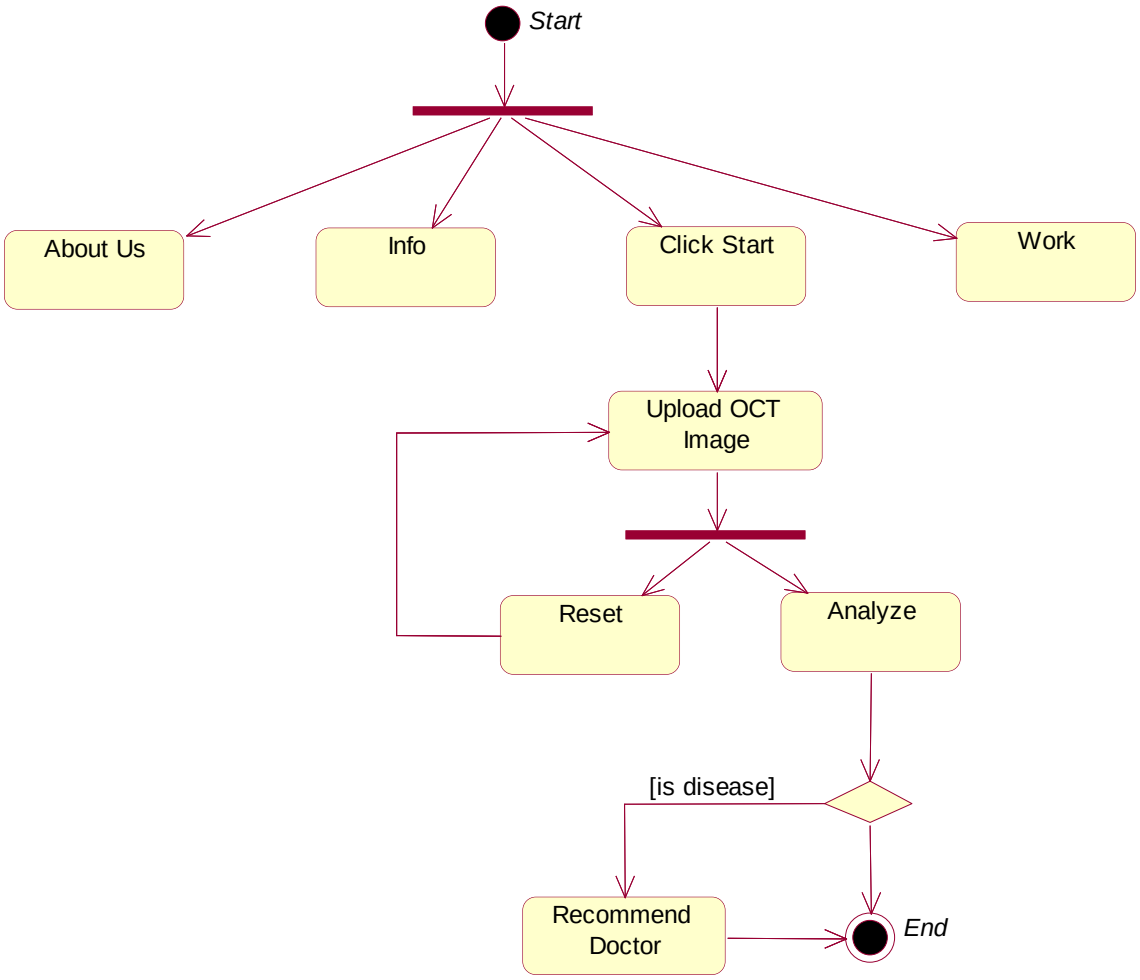


Figure 4.4 Activity diagram

4.2.3 SEQUENCE DIAGRAM

A sequence diagram is a graphical view of a scenario that shows object interaction in a time- based sequence, what happens first, what happens next. Sequence diagrams establish the roles of objects and help provide essential information to determine class responsibilities and interface.

This type of diagram is best used during early analysis phases because they are simple and easy to comprehend. A sequence diagram has two dimensions: typically, vertical placement represents time and horizontal placement represents different objects.

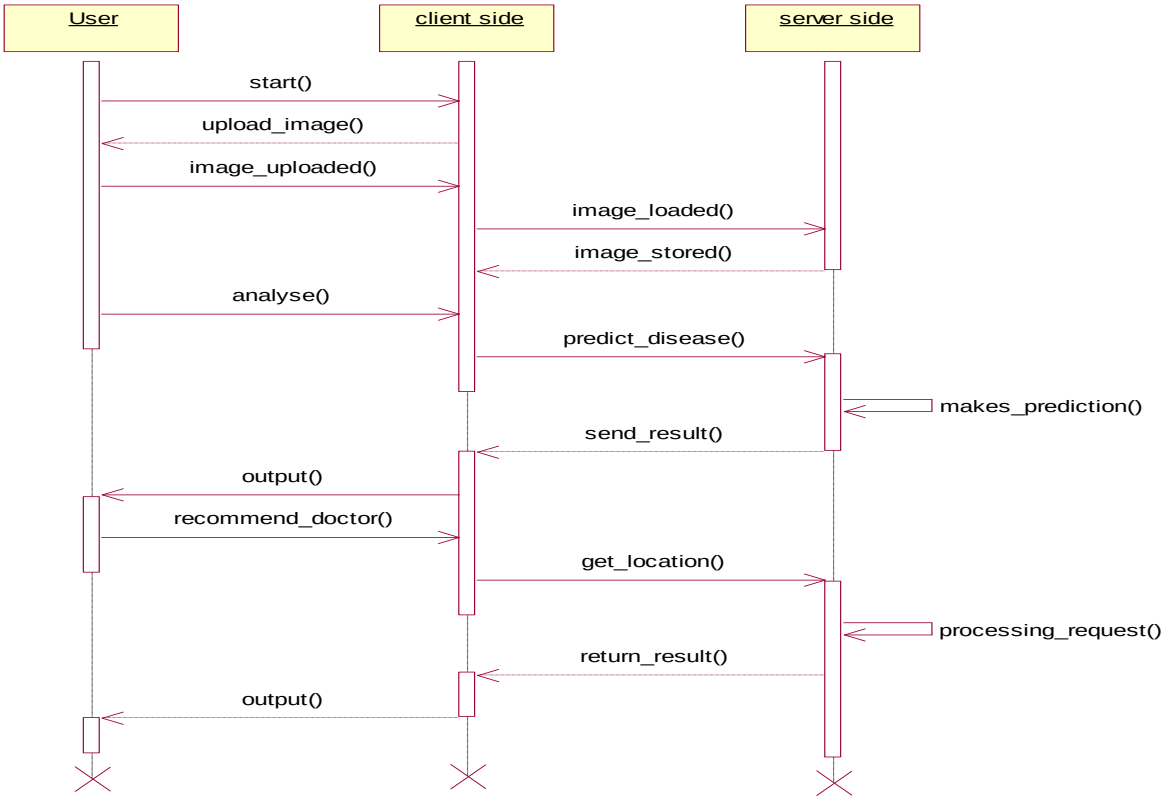


Figure 4.5 Sequence Diagram

4.2.4 DATA FLOW DIAGRAM

A Data Flow Diagram (DFD) is a structured analysis and design tool that can be used for flowcharting. A DFD is a network that describes the flow of data and the processes that change or transform the data throughout a system. This network is constructed by using a set of symbols that do not imply any physical implementation. It is the starting point of the design phase that functionally decomposes the requirements specifications down to the lowest level of detail. DFD can be considered to an abstraction of the logic of

information-oriented or a process-oriented system flow-chart.

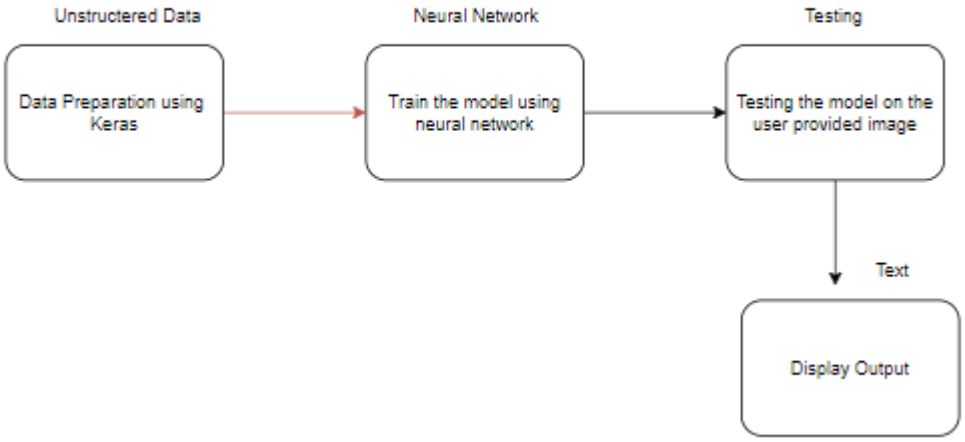


Figure 4.6 Data Flow Diagram

CHAPTER 5

PROJECT IMPLEMENTATION AND OUPUT SCREENS

5.1 SCREENSHOTS

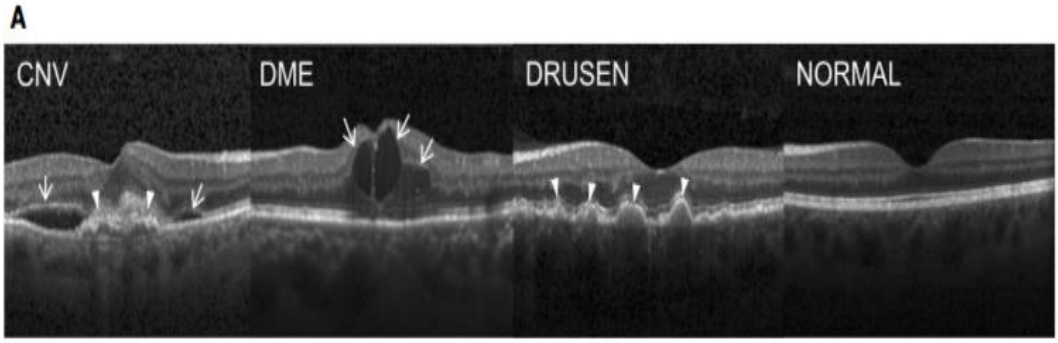


Figure 5.1 Datasets use for training purpose

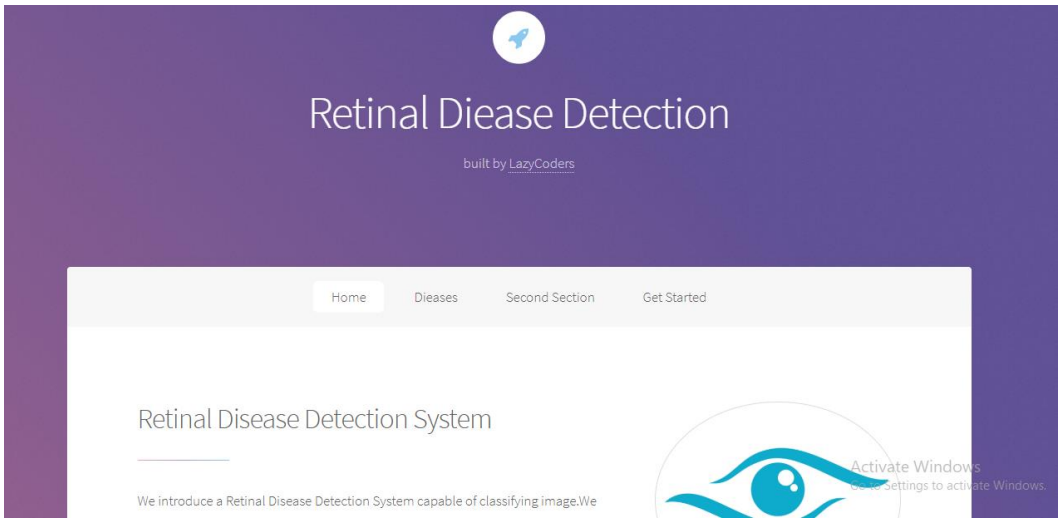
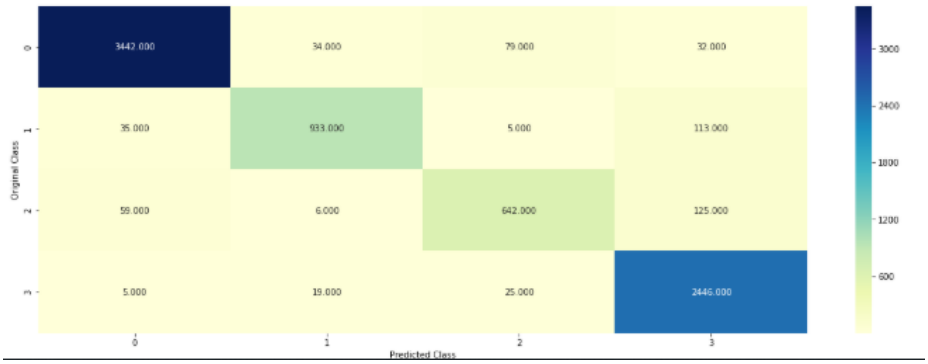


Figure 5.2 User Interface

```
plot_confusion_matrix(actual_class,pred_class)
```

```
-------------------- Confusion matrix --------------------
```



5.3 Confusion Matrix

5.2 IMPLEMENTATION OF SYSTEM

5.2.1 IMPORTANT CODING

```
def conv_layer():
model=Sequential()
model.add(Conv2D(64,(5,5),activation='relu',input_shape=(224,224,3),kernel_initializer='he_normal'))
model.add(Conv2D(64,(5,5),activation='relu',padding='same',kernel_initializer='he_normal'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Conv2D(32,(5,5),activation='relu',padding='same',kernel_initializer='he_normal'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Conv2D(32,5,activation='relu',padding='same',kernel_initializer='he_normal'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Conv2D(16,(5,5),activation='relu',padding='same',kernel_initializer='he_normal'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
model.add(Dense(64,activation='relu',kernel_initializer='he_normal'))
```

```
model.add(Dense(32,activation='relu',kernel_initializer='he_normal'))
model.add(Dense(4,activation='softmax'))
return model

model = conv_layer()
model.compile(optimizer = 'adam', loss = 'categorical_crossentropy',metrics = ['accuracy'])
```

CHAPTER 6

TESTING

Testing documentation involves the documentation of artefacts that should be developed before or during the testing of Software. Documentation for software testing helps in estimating the testing effort required, test coverage, requirement tracking/tracing, etc. This section describes some of the commonly used documented artefacts related to software testing. Testing allows you to ensure your application works the way you think it does, especially as your codebase changes over time. If you have good tests, you can refactor and rewrite code with confidence. Tests are also the most concrete form of documentation of expected behaviour, since other developers can figure out how to use your code by reading the tests.

OBJECTIVE OF TESTING

- Testing is a process of executing a program with intend of finding an error.
- A good test case is one that has a high probability of finding an undiscovered error.

These objectives imply a dramatic change in view point. They move counter to the commonly held view that a successful test is one in which no error is found. Our objective is to design test that systematically uncovered different classes of errors and to do so with a minimum amount of time and effort

6.1 TESTING STRATEGY ADOPTED

Various software-testing strategies have been proposed so far. All provide a template for testing. Things that are common and important in these strategies are: Testing begins at the module level and works "outward": tests which are carried out are done at the module level where major functionality is tested and then it works towards the integration of the entire system.

Different testing techniques are appropriate at different point of time: Under different circumstances, different testing methodologies are to be used which will be the decisive factor for software robustness and scalability. The developer of the software conducts testing and if the project is big then there is a testing team: All programmers should test and verify that their results are according to the specification given to them while coding. In cases where programs are big enough or collective effort is involved for coding, responsibilities for testing lies with the team as a whole.

There are many strategies that a project can adapt depending on the context and some of them are:

- Dynamic and heuristic approaches
- Consultative approaches
- The model-based approach that uses statistical information about failure rates.
- Approaches based on risk-based testing where the entire development takes place based on the risk.
- Methodical approach, which is based on failures.
- The standard-compliant approach specified by industry-specific standards

6.2 TEST PLAN

Before adopting and testing strategy we have to know what is testing is and what strategy should adopt for testing. So, Test strategy is an outline that describes the testing approach of the software development cycle. It is created to inform project managers, testers, and developers about some key issues of the testing process. This includes the testing objective, methods of testing new functions, total time and resources required for the project, and the testing environment. Test strategies describe how the product risks of the stakeholders are mitigated at the test-level, which types of test are to be performed, and which entry and exit criteria apply. They are created based on development design documents. System design documents are primarily used and occasionally, conceptual design documents may be referred to. Design documents describe the functionality of the software to be

enabled in the upcoming release. For every stage of development design, a corresponding test strategy should be created to test the new feature sets.

6.3 FEATURES TO BE TESTED

Software Testing

- Check whether the registration module Working
- Whether training data is well prepared or not.
- Whether we are using the best accuracy neural network.
- Test efficiency of your algorithm by test in it with other algorithms also.

6.4 SYSTEM TESTING

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black-box testing, and as such, should require no knowledge of the inner design of the code or logic. In this testing, there is no need to know about technology and language.

As a rule, system testing takes, as its input, all of the "integrated" software components that have passed integration testing and also the software system itself integrated with any applicable hardware system. The purpose of integration testing is to detect any inconsistencies between the software units that are integrated (called assemblages) or between any of the assemblages and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "inter- assemblages" and also within the system as a whole.

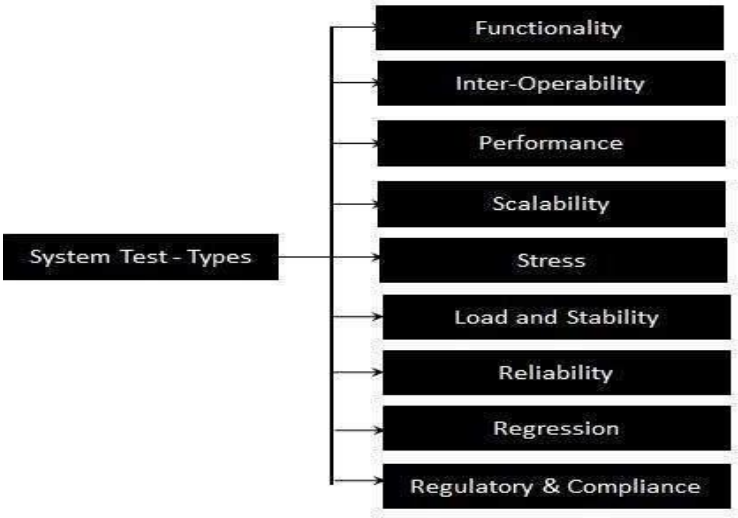


Figure 6.1 System testing diagram

6.5 UNIT TESTING

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. It is often done by the programmer to test that the unit he/she has implemented is producing expected output against given input. The following are the tests that are performed during the unit testing:

- **Registration Modules: -** To check whether the all validation is applied or not.
- **Upload Module: -** To check whether the image will properly be uploaded or not in the proper format.
- **Predict Disease : -** To check whether it will predict the result from the four class.
- **Boundary Conditions: -** It is observed that much software fails at boundary conditions. That's why boundary conditions are tested to ensure that the program is properly working at its boundary conditions.

6.6 FUNCTIONAL TESTING

Functional testing is the testing to ensure that the specified functionality required in the system requirements works. It falls under the class of black-box testing.

Functional Testing is a testing technique that is used to test the features of the system or Software, should cover all the scenarios including failure paths and boundary cases. React forces us to build everything as "components."

6.7 TEST CASES

Purpose of Test Case:

This feedback form, enquiry form, and map Test Case Document identifies all conditions to be implemented within the testing scenario. These conditions are mandatory for an acceptable and successful implementation of the function of different modules.

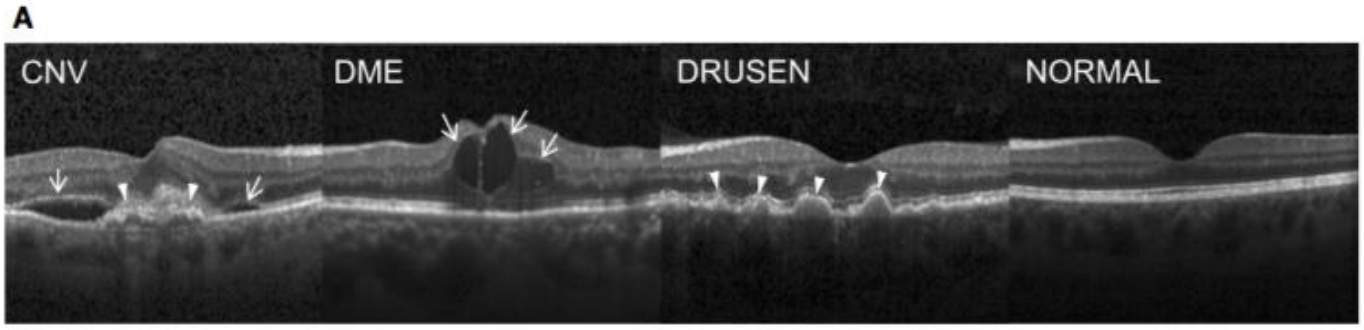
Use case no.	Test Cases	Description	Test Result
T-001	User Registration	Registration of the user helps us to provide the personal experience to the patient	Successful
T-002	Password Encryption	Password Encryption module will encrypt the user's password.	Successful
T-003	Upload Image	Upload Image will store used to upload the image for testing	Successful
T-004	Library check	All needed libraries present	Successful
T-005	Datasets	Enough datasets must be present	Successful

T-006	Predict Diagnosis	Diagnosis module will test on the trained model of the dataset	Successful
T-007	Save History	History of patients upload images saves in this module	Successful

6.8 Purpose

To determine the best outcomes by using different deep learning neural networks module. Tested different algorithms and given their outputs along with its accuracy.

Inputs



Expected Outputs & Pass/Fail criteria

The system will predict one of the class from the dataset and suggest the best doctor to patient.

Test Results

By applying different neural network architecture we able to achieve the highest accuracy on the testing dataset.

CHAPTER 7

FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

7.1 PERFORMANCE REQUIREMENT

The system has better speed, faster retrieval of information at a low cost.

7.2 RELIABILITY

The application will be very reliable as it will not interfere in the execution of other applications that are running simultaneously on the same system. Also it will not allow the user to access any hacked site or account.

7.3 RESPONSE TIME

Response time is the total amount of time it takes to respond to a request for service. That service can be anything from a memory fetch, to a disk IO, to a complex database query, or loading a full web page. Ignoring transmission time for a moment, the response time is the sum of the service time and wait time.

7.4 ROBUSTNESS

Robustness is the ability of a computer system to cope with errors during execution. Robustness can also be defined as the ability of an algorithm to continue operating despite abnormalities in input, calculations, etc. Robustness can encompass many areas of computer science, such as robust programming, robust machine learning, and Robust Security Network form, the more robust the software. Formal techniques, such as fuzz testing, are essential to showing robustness since this type of testing involves invalid or unexpected inputs.

7.5 MAINTABLITY

If The application is easy to maintain. And regular updates will be provided.

7.6 PORTABLITY:

It supports all operating system that consist a web browser.

7.7 SECURITY

Our system provides the security user by encryptng the passwords and information encrypted

7.8 STABILITY

Must be stable in all circumstance and have the strength the handle a large load

7.9 TESTING

All test cases must run properly with any fault in the result.

7.10 FAULT TOLERANCE

Fault tolerance is the property that enables a system to continue operating properly in the event of the failure of (or one or more faults within) some of its components. All the

CHAPTER 8

CONCLUSION

To summarize, this project report we propose a deep learning-based on CNN models that can differentiate normal and three different classes of retina disease and provide doctor recommendations. To implement this concept, we also developed a website with personal experience on this AI platform. In its present state, the website can help doctors and patients who wish to ascertain a patient's status and receive better experience to the patient. We only predict the disease based on the previous data so, Therefore, patients decision on treatment should be based not solely on the results from the AI classifier but, most importantly, on clinical judgement. our software will suggest the patient seek medical help if the active retinal disease is suspected. The definite diagnosis and treatment should be performed by a retinal specialist based on clinical evidence and experience.

8.1 FUTURE ENHANCEMENT

If we talk about future enhancement we can try to personal assistant to the user so that the assistant give some suggestion to the patient every week and we also try to use online deep learning so that our model will improve on the user data and make good predictions. We also try to switch our platform on aws to store the user pieces of information and try to analyse so will be better experienced to patient.

Appendix A:

GLOSSARY

SR NO.	WORDS	MEANINGS
1	Authentication	Process of establishing who you are.
2	Authorization	Permission to access non-public information or use equipment that is either fully or partially restricted. Process of establishing what you can do.
3	Browser	A software tool used to read electronic documents. Mosaic, Netscape and Lynx are the most popular browsers.
4	Client/server	A relationship in which client software obtains services from a server on behalf of a person.
5	Crash	A computer system is said to crash when it stops working for some reason and must be restarted.
6	Database	A collection of interrelated data values that may be integrated permanently into a single connected structure or integrated temporarily for each interrogation, known as a query.
7	Database management system	A systematic approach to storing, updating, securing and retrieving information stored as data items, usually in the form of records in one or more files.
8	Error message	A message that reports the detection of an error.
9	Execute	To interpret computer instruction and carry out the operations specified in the instruction.
10	GUI	Graphical User Interface. Defines a format for scroll bars, buttons, menus, etc., and how they respond to the user.
11	Module	A logically self-contained and discrete part of a larger computer program.
12	SQL	Structured Query Language. ANSI standard data manipulation language used in most relational data base systems. A language for requesting data

		from a relational database.
13	User	Anyone who uses a computer connected to the Internet.
14	Username	Account name or user ID.
15	Utility	A specialized program that performs a frequently required everyday task such as sorting, report program generation, or file updating.

x

Appendix B:

REFERENCES

1. Christiansen EM, Yang SJ, Ando DM. et al. In silico labelling: Predicting fluorescent labels in unlabeled images. Cell. 2018;173:792–803.
2. Kermany DS, Goldbaum M, Cai W. et al. Identifying medical diagnoses and treatable diseases by image-based deep learning. Cell. 2018;172:1122–31
3. Kocur, I., Resnikoff, S.. Visual impairment and blindness in europe and their prevention.
4. Mookiah, M.R.K., Acharya, U.R., Chua, C.K., Lim, C.M., Ng, E., Laude, A.. Computer-aided diagnosis of retinal diseases.
5. Cun, Y.L., Boser, B., Denker, J.S., Howard, R.E., Habbard, W., Jackel, L.D., et al. Advances in neural information processing systems.

BIBLIOGRAPHY

[1] Programming Collective Intelligence.

[2] Machine Learning for Hackers.

[3] Machine Learning by Tom M Mitchell.

[4] The Elements of Statistical Learning.

[5] Introduction to Machine learning by Andrew NG

ABBREVATION

[1] ML :Machine Learning

[2] AI: Artificial Intelligence

[3] CNN: Convolutional Neural Network

[4] OCT: Optical Coherence tomography

[5] CNV: Choroidal Neovascularization

[6] DME: diabetic macular edema