Diabetic Retinopathy Detection

(Web Based Application)

Final Year Project Report

Group Members

Ridha Shafiq

Syeda Saman Fatima

Fatimah Ahmad

Yalmaz Khan

Supervisor

Dr. Naima Iltaf

"In Partial Fulfillment of the Requirements for the Degree

Bachelor of Engineering in Software Engineering (BESE)"

Military College of Signals

National University of Sciences and Technology

Islamabad, Pakistan

(May 2023)

In the name of ALLAH, the Most Benevolent, the Most Courteous

DECLARATION

We affirm that this project report titled ***"Retina Doc - Web-Based Application for Diabetic Retinopathy Detection"*** presented to the ***"Department of Computer Software Engineering"*** is an original work undertaken by us under the supervision of ***"Dr. Naima Iltaf"***. We assure you that no part of this report has been copied from any source without appropriate citations. Furthermore, this project has been submitted to fulfill the requirements for the degree of Bachelor of Computer Science

Team Members	**Signature**
Ridha Shafiq	_______________
Syeda Saman Fatima	_______________
Fatimah Ahmad	_______________
Yalmaz Khan	_______________

Supervisor	**Signature**
Dr Naima Iltaf	_______________

Date:

____

Place:

____

DEDICATION

I dedicate this project to my parents, whose unwavering support and encouragement have been my driving force. To my professors, whose guidance and expertise have molded me into a capable individual. To my friends, who have kept me grounded and provided me with endless motivation. And finally, to all the individuals whose participation in this project has made it possible. Without your contributions, this work would not have been possible.

May this project serve as a testament to the power of perseverance, hard work, and the pursuit of knowledge.

ACKNOWLEDGEMENTS

"Allah Subhan'Wa'Tala is the sole guidance in all domains"

I would like to express my deep appreciation to all those who have contributed to the successful completion of our final year project on the Diabetic Retinopathy Web Application.

Firstly, we extend our gratitude to our supervisor for providing us with invaluable guidance and mentorship throughout the project, as well as their constructive feedback that helped shape the project and ensured its success. We would also like to thank the National University of Sciences and Technology (NUST) faculty members and staff for their support and for providing us with the necessary resources and facilities to conduct our research and complete this project.

Our sincere thanks go to the doctors and healthcare professionals who generously provided us with the diabetic retinopathy dataset from a renowned Al-Shifa Eye Trust Hospital in Pakistan. This dataset was essential in developing our machine learning model and helped us achieve accurate diagnosis and treatment of diabetic retinopathy. Additionally, we thank our classmates and friends for their valuable feedback and suggestions that improved the project's quality and ensured its success.

Finally, we would like to express our gratitude to our families for their unwavering support and encouragement throughout our academic journey. Their love and support have been vital to our success. Once again, we extend our heartfelt thanks and appreciation to everyone who has contributed to the success of this project.

Plagiarism Check Certificate (Turnitin Report)

This Thesis has been checked for plagiarism and has 15% similarity index. The Turnitin report endorsed by Supervisor is attached at the end of this thesis.

Ridha Shafiq

00000285925

Syeda Saman Fatima

00000285424

Fatimah Ahmad

00000287689

Yalmaz Khan

000003254083

Signature of Supervisor: *Dr. Naima Iltaf*

ABSTRACT

The Diabetic Retinopathy Detection Application is a cutting-edge, web-based decision-support system that leverages automated screening to detect DR, which is a significant contributor to early blindness in diabetic patients. The system streamlines the screening process and reduces ophthalmologists' workload by automating the report generation process. The project's scope includes identifying key challenges for DR detection, researching recognition techniques, and implementing the proposed method. The project also aims to transfer important diagnostic, teaching, and research technologies, establish a medical image processing research lab in future, and form an innovative incubation center to address industry-related problems. Ultimately, the Diabetic Retinopathy Detection Application (Retina Doc) aims to reduce the risk of blindness among diabetic patients, especially in remote areas deprived of high-end machines that can detect DR, and save individuals from the consequences of DR.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Chapter 1

INTRODUCTION

1.1 PURPOSE

Diabetes mellitus (DM) is a chronic disease that affects a large population globally and is known to be the fourth leading cause of death. According to the International Diabetes Federation (IDF), the current global prevalence of DM is 336 million, with an estimated increase of up to 7.7% by 2030 [1, 2]. Diabetic retinopathy (DR), which is a complication of DM, is a primary cause of irreversible blindness in adults worldwide [3]. The research project aims to develop an innovative web-based screening system that can identify various stages of DR from color fundus images. The proposed system is intended to be a medical health tool that provides ophthalmologists with a reliable second opinion to diagnose DR accurately.

1.2 PROBLEM STATEMENT

Diabetic retinopathy is a severe eye condition that can result in blindness if not treated promptly. It is prevalent among individuals with diabetes and is caused by harm to the blood vessels in the retina. To prevent vision loss, it is critical to identify and treat the condition early. However, there is still a significant lack of reliable and user-friendly platforms for the early detection of diabetic retinopathy that can analyze retinal images uploaded by users and accurately identify any signs of the condition.

Therefore, the problem statement for this project is to develop a web-based application for the detection of diabetic retinopathy that can offer a dependable and user-friendly interface for early detection and treatment of the condition.

1.3 PROPOSED SOLUTION

Currently, significant progress has been made on the development of a web-based application for the early detection and management of diabetic retinopathy. The application is designed to be accessible to healthcare professionals and patients alike, allowing them to upload retinal images and receive a timely and efficient diagnosis. Moving forward, the project aims to provide guidance and resources for those diagnosed with diabetic retinopathy, such as specialist referrals and information on treatment options.

Ultimately, the objective of the web application is to reduce the risk of vision loss and enhance the quality of life for those impacted by this condition.

1.4 WORKING PRINCIPLE

A diabetic retinopathy detection web-based app works through the following steps:

- **User uploads retinal image:** The user uploads a retinal image of their eye to the web-based app. The image can be uploaded from a computer or mobile device.
- **Image pre-processing:** The app performs pre-processing on the uploaded image to ensure that it is of suitable quality and size for analysis. This may include resizing, normalization, and noise reduction.
- **Image analysis:** The application employs machine learning algorithms, such as convolutional neural networks (CNNs), to examine the retinal image and identify indications of diabetic retinopathy, such as microaneurysms, hemorrhages, and exudates.
- **Diagnosis:** Based on the analysis results, the app provides a diagnosis of the user's condition, indicating whether they have signs of diabetic retinopathy.

The automated examination of retinal images provided by the web-based application for the detection of diabetic retinopathy through image processing techniques and machine learning algorithms enables users to detect the signs of diabetic retinopathy early, take preventive measures, and improve their quality of life.

1.5 PRODUCT SCOPE

Retinal fundus image analysis (RFIA) is a widely adopted screening method for diabetic retinopathy (DR) that has helped to address the risk of blindness among diabetic patients [3, 4]. The use of RFIA screening programs has proven to be an effective means of addressing this significant visual impairment problem [5-8]. Early blindness can result from DR, which is one of the most prevalent conditions among diabetic patients. However, current methods of DR detection require the expertise of an experienced ophthalmologist and a careful examination of multiple retina scans, which is a time-consuming process. Even lesions that indicate DR, such as micro-aneurysms (red lesions), hemorrhages, and exudates (bright lesions), are not always visible and require skill to identify [9, 10]. Figure 1 provides examples of DR-related lesions in retinal fundus images. A clinical study [11] found that DR can be classified into two classes: non-proliferative (NPDR) and proliferative DR.

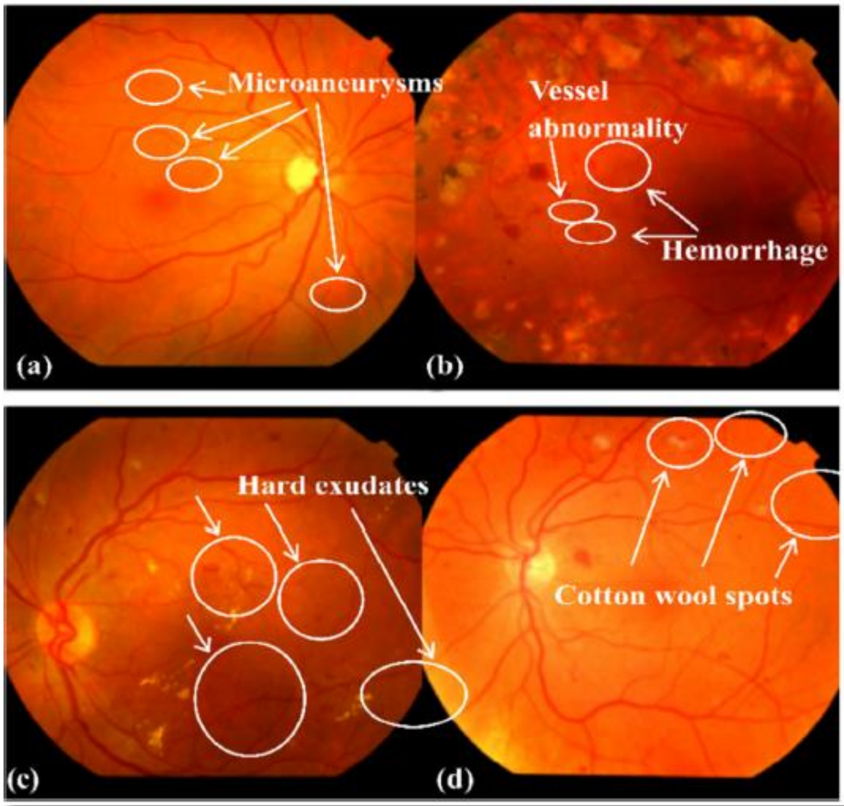


Figure 1. A colored fundus sample with retina complications:

(a) no lesions, (b) MAs, (c) HEMs, (d) vessel irregularity, (e) EXs, and (f) CWS

The proposed product aims to automate and expedite the process of detecting diabetic retinopathy, which currently leads to approximately 75 people going blind every day. By allowing patients to upload their retina scans, the system provides a report detailing the level of damage, affected area, and potential treatment options, thereby reducing the need for unnecessary resource utilization and minimizing the consequences of DR. The system is also beneficial for individuals residing in remote areas that lack high-end machines capable of detecting DR. Additionally, the product streamlines the workflow for doctors by replacing the manual process with an automated one, resulting in faster report generation.

The project has several categories of objectives, which include research, academic, industrial, human resource development, and infrastructure objectives. These categories will be elaborated on in the following sections.

1.6 OBJECTIVES

1.6.1 Research Objectives

i) Investigate and document the primary challenges associated with identifying the type and severity of diabetic retinopathy.

ii) Utilize a color fundus camera to accurately classify the type of diabetic retinopathy.

iii) Research and evaluate various methods and algorithms for the automatic recognition of diabetic retinopathy.

iv) The aim is to create a web-based decision-support system for detecting diabetic retinopathy that utilizes the proposed method.

1.6.2 Academic Objectives

i) Exhibiting a prototype of the system at the University could serve as a motivating factor and a source of inspiration for students, thus facilitating research and development.

ii) The project aims to facilitate the transfer of important medical technologies, including diagnostic, teaching, and research, from the hospital to MCS-NUST.

iii) Providing formal training sessions, utilizing video conferencing, and establishing long-term relationships, to transfer technical and professional skills to students and researchers, which could prove to be highly beneficial for them.

1.6.3 Industrial Objectives

i) Our exploration of current computational image processing and machine learning techniques can assist in the implementation of efficient medical projects in the software and medical informatics industries.

ii) The proposed research project aims to establish a medical image processing research lab at MCS-NUST, equipped with the ability to undertake and carry out global projects.

iii) This project's development will contribute to the creation of an innovative incubation center dedicated to addressing industry-related challenges.

iv) The outcome of the proposed research project can be marketed to ophthalmology departments globally, providing a potential source of revenue.

1.7 DELIVERABLES

The deliverables for a diabetic retinopathy detection web application can be categorized into different areas such as functional, technical, and design aspects. Below are some of the possible deliverables for each of these areas:

1.7.1 Functional Deliverables

i) **User Registration and Login System:** This allows users to create an account, log in, and access the application's features.

ii) **Retinal Image Uploading:** The web application should enable users to upload retinal images in various formats, such as JPEG or PNG.

iii) **Image Analysis and Detection:** To detect indications of diabetic retinopathy in retinal images, the program must incorporate algorithms capable of analyzing the uploaded images for the presence of diabetic retinopathy markers, such as microaneurysms, hemorrhages, and exudates.

iv) **Diagnosis:** The application should provide a diagnosis (Diseased/Non-Diseased)

1.7.2 Technical Deliverables

i) **Database for Authentication:** A database should be created to store users' credentials.

ii) **Web Application Framework:** The web application should be built using a suitable web application framework Django.

iii) **Image Processing Library:** The application should use an image processing library, such as OpenCV, keras, image data generator etc., to process and analyze retinal images.

iv) **Machine Learning Algorithms:** To detect signs of diabetic retinopathy in retinal images, the program should utilize machine learning techniques like convolutional neural networks (CNNs)

v) **Hosting:** The web application should be hosted on a reliable web hosting service (localhost).

1.7.3 Design Deliverables

i) **User Interface Design:** The web application should have a user-friendly interface that is easy to navigate and use.

ii) **Branding and Identity:** The web application should have a unique and recognizable brand identity that reflects the purpose of the application.

iii) **Visual Design:** The web application should have a visually appealing design that is consistent with the brand identity.

1.7.4 Future Deliverables

i) **Feedback:** The application will give feedback to the user about their condition based on analysis results.

ii) **Referral and Treatment Options:** The application should provide guidance and resources for users who are diagnosed with diabetic retinopathy, including referral to a specialist and information on treatment options.

iii) **User Management System:** The application should have a system to manage users' data and access rights.

iv) **Database:** A database should be created to store users' data, retinal images, and analysis results.

v) **Deployment:** In future the web application should be deployed using a web server like Apache or Nginx.

vi) **Responsive Design:** The web application should be designed to be responsive, ensuring that it works on different devices and screen sizes.

Overall, the deliverables for our application "Retina Doc" focuses on providing a reliable, user-friendly, and efficient platform for early detection and management of diabetic retinopathy.

1.8 DATA DESIGN

After gathering a substantial dataset of retinography images, consisting of 1300 DR retinography images and 1300 regular retinography images, from both public and private sources such as Al-Shifa Eye Trust Hospital and online-kaggle, we utilized a pre-trained deep learning model and fine-tuned it on this dataset to improve accuracy. This method allowed for faster and more efficient training, better generalization, and improved performance on new tasks.

To ensure accuracy, the dataset was curated by an experienced ophthalmologist who manually distinguished between DR and regular fundus photos from various sources. Furthermore, for analysis purposes, all images were resized to 224 × 224 pixels.

Table 1. Retinal Fundus Image Dataset

Title	Name	DR	Non-DR	Size	Fundus Images
[Public]	Kaggle	500	500	(1152 x 1500) pixels	1000
[Private]	Al-Shifa Eye Trust	800	800	(1125 x 1264) pixels	1600
		1300	1300		2600

1.9 RELEVANT SUSTAINABLE DEVELOPMENT GOALS

Our project is mapped on following SDGs:

a) **SDG1 (No Poverty):** The use of a private dataset from Pakistan's famous eye trust hospital targeting the Asian ethnicity in this web application can contribute to the SDG of No Poverty by improving healthcare outcomes and addressing health disparities. This can help reduce poverty by ensuring that people can live healthy lives and access essential healthcare services, which can prevent the onset of poverty due to illness or disease. By providing better access to medical care and improving the outcomes of treatment, people with diabetes in rural areas of Pakistan can avoid the complications of the disease, which can lead to poverty due to lost income and increased healthcare costs.

b) **SDG 3 (Good Health and Well-Being):** Attaining sustainable development requires ensuring the well-being and promoting healthy lives for everyone, regardless of age. Health is a significant aspect of the SDGs, and the declaration highlights the importance of achieving universal health coverage and quality healthcare to achieve the overall health goal. The SDGs prioritize ensuring that no one is excluded or left behind in accessing these services.

c) **SDG 4 (Quality Education):** Presenting a system prototype at the university can inspire and motivate students, driving research and development forward. The prototype can serve as a catalyst for further innovation and progress.

d) **SDG8(Decent Work and Economic Growth):** By utilizing this dataset, it is possible to facilitate the creation of innovative technologies that can enhance healthcare delivery in rural parts of Pakistan and other low-income nations. This, in turn, can stimulate economic growth and employment opportunities, thus alleviating poverty and elevating the living standards of people.

e) **SDG 9 (Industry, innovation, and Infrastructure):** The project is a marketable product that can be sold to ophthalmology departments worldwide as a useful tool, particularly in rural areas where access to fundus image capture devices may be limited.

1.10 INTENDED AUDIENCE AND READING SUGGESTIONS

This project is used for detecting and classifying Diabetic Retinopathy. It has been implemented under the guidance of college professors. This project is useful for ophthalmologists as well as patients with diabetes. It can also be used as a reference by students for research purposes.

1.11 GENERAL FORMAT OF THESIS

The following is the generic format for this documentation.

- **Chapter 1:** Introduction
- **Chapter 2:** Literature Review

- **Chapter 3:** System Features
- **Chapter 4:** Methodology
- **Chapter 5:** Detailed Design and Architecture
- **Chapter 6:** Other Non-Functional Requirements
- **Chapter 7:** Implementation
- **Chapter 8:** Testing, Results and Discussion
- **Chapter 9:** Interface and User Manual
- **Chapter 10:** Conclusion and Future Work
- **Chapter 11:** References and Works Cited

1.12 STYLES

This section describes the standards followed while writing this document.

1.12.1 Typeface

The document is formatted with 1.5 line spacing, TNR font of size 12, and justified text. The first line of each paragraph is indented, and a single line space separates each paragraph.

1.12.2 Margins

The left margin is set to 1.5 inches and the top, right, and bottom margins to 1.2 inches.

1.12.3 Headings

In this document, Chapter Number is in 16 TNR font with italicized and bold text that is right-justified, and the first letter capitalized. The Chapter Heading is in 16

TNR font with bold text that is all capitalized and centered. All subsequent headings are left-aligned, with text starting from the next line and an indentation.

The First Level Heading is in 14 TNR font with bold text that is all capitalized. The Second Level Heading is in 12 TNR font with bold text, and the first letter of each main word is capitalized. The Third Level Heading is in 12 TNR font with bold text, and only the first letter of the first word is capitalized.

1.12.4 Tables and Figures

Each table has a title and table number located at the top of the table (e.g., "Table 1. Abc"). Titles of figures are placed below the figures, along with the figure number (e.g., "Figure 1. Xyz").

1.13 REFERENCES

All the references including journals, books, and references from the internet used for this project have been cited in Chapter 11 of this thesis.

Chapter 2

Literature Review

Chapter 2

LITERATURE REVIEW

A new tool is made by studying the different research papers written on diabetic retinopathy detection. Literature review is an important step for development of an idea to a new product. Likewise, for the making of new tool related to the Retinopathy detection, a detailed study of the research papers written for this is necessary. Our research is categorized into the following.

- Industrial Background
- Existing solutions and their drawbacks
- Research Papers

2.1 INDUSTRIAL BACKGROUND

In Pakistan, diabetic retinopathy is a significant public health concern due to the increasing prevalence of diabetes in the country. According to the International Diabetes Federation, Pakistan has the seventh-highest prevalence of diabetes in the world, with approximately 19.4 million adults living with the condition.

The industrial background for diabetic retinopathy detection in Pakistan involves the use of technology and healthcare to address this issue. The healthcare industry has been actively developing and implementing screening programs and tools to detect diabetic retinopathy early, with the goal of preventing vision loss.

In recent years, there has been a growing interest in the use of artificial intelligence (AI) and machine learning (ML) for diabetic retinopathy detection in Pakistan. Several hospitals and healthcare institutions in the country have started implementing AI-based diabetic retinopathy screening programs.

For example, the Aga Khan University Hospital in Karachi has implemented an AI-based diabetic retinopathy screening program that uses a retinal camera and machine learning algorithms to detect the condition. Similarly, the Punjab Institute of Cardiology in Lahore has implemented an AI-based diabetic retinopathy screening program that uses a fundus camera and deep learning algorithms.

Additionally, several companies in Pakistan have developed and marketed diabetic retinopathy screening devices, such as retinal cameras and OCT scanners. These devices are used by ophthalmologists and optometrists to capture images of the retina and detect signs of diabetic retinopathy.

Furthermore, the government of Pakistan has recognized the importance of diabetic retinopathy detection and management and has implemented several initiatives to address the issue. The National Eye Health Program, launched by the Ministry of National Health Services, Regulations, and Coordination in 2019, aims to provide free eye screening services and treatment to individuals with diabetes.

Overall, the industrial background for diabetic retinopathy detection in Pakistan involves a combination of technology, healthcare, and research efforts aimed at improving the early detection and management of this condition. The use of AI and ML algorithms has the potential to revolutionize diabetic retinopathy detection and improve patient outcomes in the country.

2.2 EXISTING SOLUTIONS AND THEIR DRAWBACKS

There are several existing solutions for diabetic retinopathy detection systems in Pakistan, including traditional screening methods and more advanced technology-based solutions. However, each of these solutions has its own advantages and drawbacks.

Traditional methods of diabetic retinopathy detection in Pakistan involve dilated fundus examination by an ophthalmologist, which can be time-consuming and requires specialized equipment and trained personnel. Another traditional method is the use of retinal photography, which involves capturing images of the retina and analyzing them for signs of diabetic retinopathy. However, this method also requires specialized equipment and trained personnel.

Recently, there has been an increasing interest in the use of technology-based solutions for diabetic retinopathy detection in Pakistan, including artificial intelligence (AI) and machine learning (ML) algorithms. Several hospitals and healthcare institutions in Pakistan have implemented AI-based diabetic retinopathy screening programs using retinal cameras and machine learning algorithms. These programs can analyze images of the retina and detect signs of diabetic retinopathy with high accuracy.

One of the drawbacks of these AI-based screening programs is the high cost of implementation and maintenance. Additionally, there is a shortage of trained personnel to operate the screening equipment and analyze the results. The lack of awareness among individuals with diabetes about the importance of regular screening for diabetic retinopathy is another significant challenge.

Another drawback of existing solutions for diabetic retinopathy detection systems in Pakistan is the lack of a coordinated and integrated approach due to lack of proper ethnically incline Endo-Asian Dataset. Screening programs may be implemented by different healthcare institutions or organizations, which can lead to fragmentation and inefficiencies in the overall system.

In conclusion, while there are existing solutions for diabetic retinopathy detection systems in Pakistan, there are also several challenges that need to be addressed to improve the effectiveness and accessibility of these solutions. Addressing these challenges will require a coordinated and integrated approach involving healthcare providers, policymakers, and individuals with diabetes to ensure that screening programs are accessible, efficient, and effective.

2.3 RESEARCH PAPERS

In the development of this project, I drew heavily on the research of Dr. Imran Qureshi, whose groundbreaking paper on Computer-Aided Detection of Hypertensive Retinopathy Using Depth-Wise Separable CNN, and Diabetic retinopathy detection and stage classification in eye fundus images using active deep learning provided critical insights that guided my approach. By integrating the findings of Dr. Imran Qureshi and other leading experts in the field, I was able to develop a project that is informed by the latest research and best practices. The two main Research Papers have been cited below:

- Qureshi, I.; Abbas, Q.; Yan, J.; Hussain, A.; Shaheed, K.; Baig, A.R. Computer-Aided Detection of Hypertensive Retinopathy Using Depth-Wise Separable CNN. Appl. Sci. 2022, 12, 12086. https://doi.org/10.3390/ app122312086

- Qureshi, I., Ma, J. & Abbas, Q. Diabetic retinopathy detection and stage classification in eye fundus images using active deep learning. *Multimed Tools Appl* 80, 11691–11721 (2021). https://doi.org/10.1007/s11042-020-10238-4

Chapter 3

System Features

Chapter 3

SYSTEM FEATURES

This section demonstrates RetinaDoc's most prominent features and explains how they can be used and the results they will give back to the user.

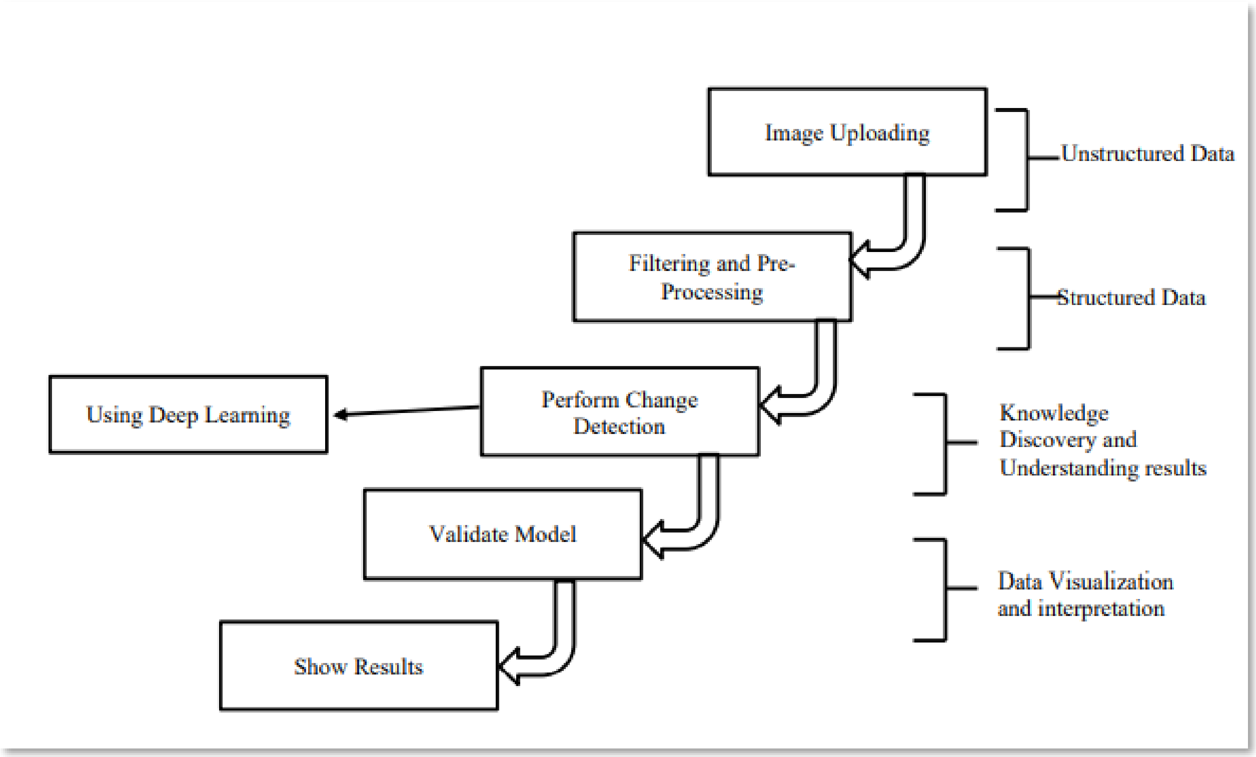


Figure 2. System Features

3.1 USER REGISTRATION AND PROFILE SETUP

i) This feature allows any user expert or typical to register themselves in our application.

ii) Any new user will be directed to the registration page where they must provide the email and password they will require to log in to the application.

iii) After the email and password is set account is successfully registered.

iv) Registration of the user is helpful for updating the database to generate the results.

3.2 USER LOGIN

i) This feature allows the registered user to login using their credentials.

ii) Once login is successful, the user is directed to a page where they can upload a retina scan.

iii) Login is only successful on giving proper credentials.

3.3 UPLOAD IMAGE OF RETINA SCAN

i) This feature allows uploading of the scanned image of retina scan via computer.

ii) The user clicks on the option of upload and uploads the image either by scanning the image onto the computer and then selecting the file or by drag and drop.

iii) Uploading of the image should be done properly by the users. The quality of the images determines the response time and the accuracy of the result obtained.

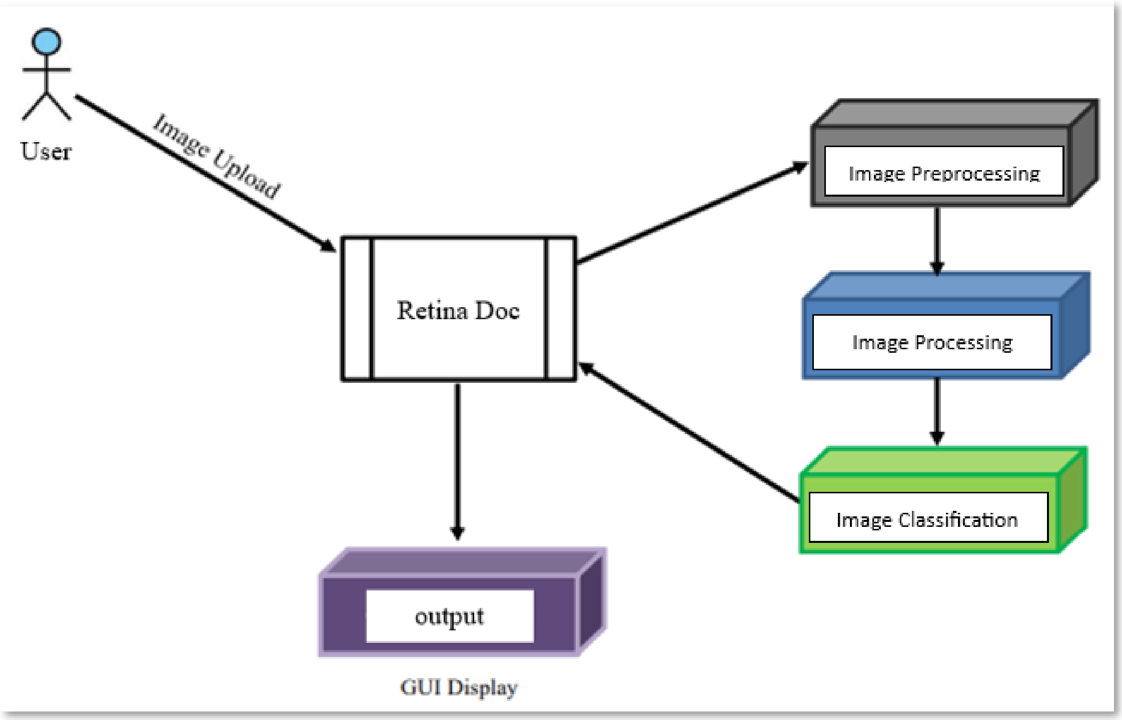


Figure 3. Sequence of Steps

3.4 DETERMINATION OF THE CLASS OF DR

i) This is the major feature of this application where the user gets the result of their retina scan and whether they have DR or not.

ii) After the user uploads the image of the retina scan, the application sends it to the saved model of the neural network and the model processes this image.

iii) The result of this functionality being the retina scan that is uploaded is classified into 2 different classes DR/Non-DR.

3.5 RESULT VISUALIZATION/ REPORT GENERATION

i) This feature enables the user to have the result converted into a report for easy understanding and quick diagnosis in case of a doctor's visit.

ii) After the class of DR is detected and the feature which is affected is identified, the report is generated which includes the afore-mentioned results stated in simpler terms for easier and thorough understanding of the results for typical users without any jargons.

iii) The user will be able to take a screenshot of the report.

Chapter 4

Methodology

Chapter 4

METHODOLOGY

4.1 WORKFLOW DIAGRAM

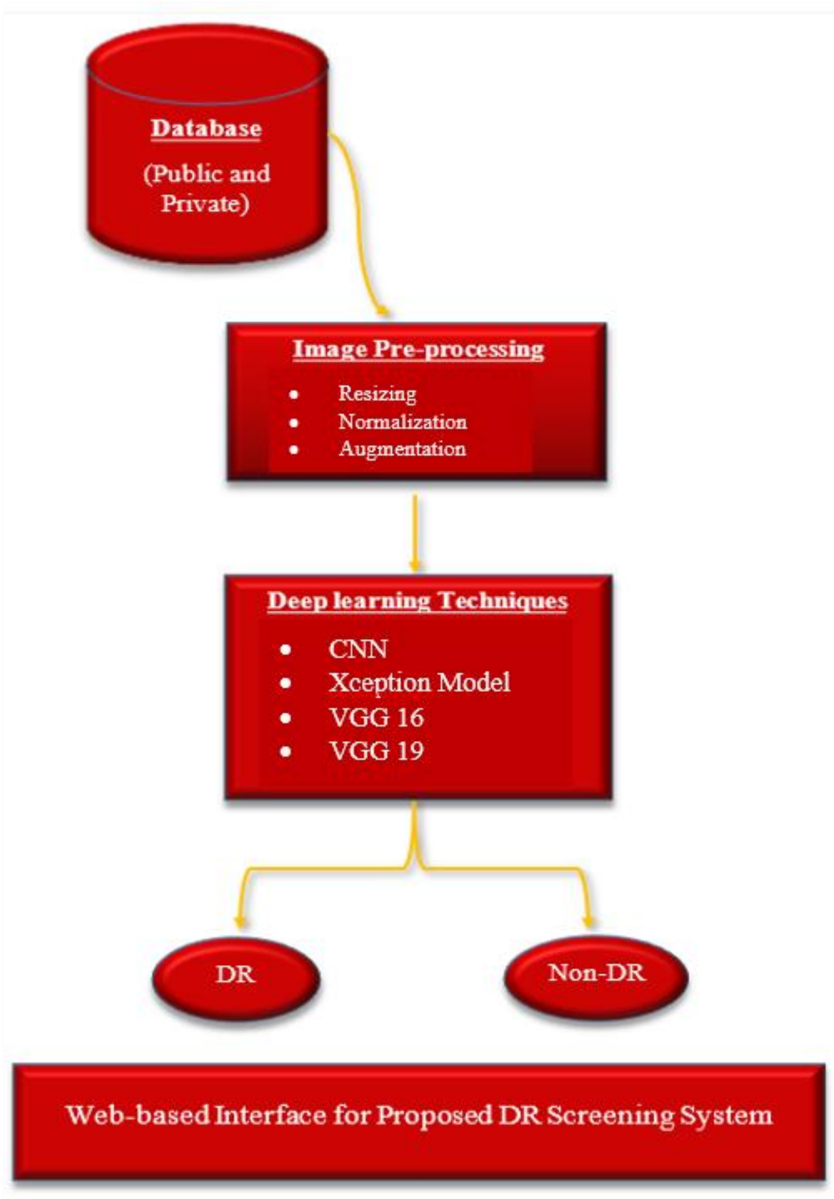


Figure 4. Workflow Diagram

4.2 OPERATING ENVIRONMENT

The software will operate with the following software components and applications:

The software being developed will be running under Mac OS 10.14.6 and/or Windows 10 operating systems. The hardware that will be running these programs will follow the specifications that appear in this document in section (4.4). To briefly state, Camera with inbuilt webcam, GPU, High Speed connection to database, and a strong Internet access.

4.3 DESIGN AND IMPLEMENTATION CONSTRAINTS

- **Memory:** For designing purposes, Google drive with a memory of about 15GB will be used for storing and loading the dataset.
- **Language requirements:** Software will be available only in the English language.
- **Programming Constraints:** Python 3 language, TensorFlow backend 1.15.0

4.4 HARDWARE REQUIREMENTS

i) A computer with a Strong Internet Access

ii) High Speed connection to Database

iii) GPU

4.5 SOFTWARE REQUIREMENTS

i) **IDE:** Anaconda 2022.10

ii) **Deep Learning Libraries:** Tensorflow, keras, ImageJ, Pandas, Numpy, Open CV

iii) **Python Framework:** CNN (vgg16 model, vgg19 model, Xception Model)

iv) **For Application Dashboard:** React, Figma

v) **Backend:** Django, sql workbench

4.6 WORKING PRINCIPLE

The project mainly works on the principles of image processing merged with machine learning algorithms. The project is divided into different modulus and every module is inter-woven with the next module. The list of modules is as under:

- Datasets and Annotations
- Dataset Training and Processing
- Feature Extraction
- Feature Classification
- GUI Presentation

4.6.1 Dataset and Annotations

An integral part of the project is the gathering of datasets (private). The dataset comprises of 2 types of retinal fundus images (diseased, and non-diseased). Our privately collected dataset from Al-Shifa Eye Trust Hospital is of main importance in our project. The images are gathered, filtered, and annotated to detect the presence of lesions in the eye such as micro-aneurysms (red lesions), hemorrhages and exudates (bright lesions) which are not visible to the naked eye.

4.6.2 Dataset Training and Processing

The prepared dataset is used as input to train DR detection models using machine learning. We have fine-tuned the private dataset that we've gathered from Al-Shifa Eye Trust Hospital over the public dataset that was previously being used to get better accuracy and performance.

4.6.3 Xception Model for Image Classification

Xception is a deep convolutional neural network (CNN) model for image classification. It was introduced by Google in 2016 and is based on the concept of depthwise separable convolutions, which reduce the number of parameters required for training while maintaining accuracy. Xception has shown superior performance on benchmark datasets compared to other CNN models, such as VGG, Inception, and ResNet. It is often used for transfer learning, where pre-trained weights from Xception are fine-tuned on a new dataset for a specific task

The main idea behind Xception Model is to use residual blocks, which allow for the training of very deep neural networks. In a traditional neural network, each layer transforms the input into a new representation, which is then passed to the next layer. In a residual block, the input is passed through a "skip connection" to the output, allowing for the original input to be preserved. This helps to prevent the vanishing gradient problem, which can occur when gradients become very small as they propagate through many layers.

For diabetic retinopathy, Xception Model can be used as a classifier to identify and classify the different stages of diabetic retinopathy based on retinal images. The Xception Model takes as input a retinal image and produces as output a prediction of the stage of diabetic retinopathy.

In the implementation of Xception Model for diabetic retinopathy, the model is typically pre-trained on a large dataset, such as ImageNet, to learn a general representation of images. Then, the pre-trained Xception model is fine-tuned on the specific diabetic retinopathy dataset, where the final layer(s) of the network are replaced or retrained to classify the different stages of diabetic retinopathy.

Overall, Xception Model is a powerful deep neural network model that can be used for various machine learning tasks, including image classification for diabetic retinopathy. Its use of residual blocks allows for the training of very deep neural networks, and its pre-training on a large dataset enables it to learn a general representation of images that can be fine-tuned on specific tasks.

4.6.4 Feature Extraction

The features are extracted based on lesions in the eye such as micro-aneurysms (red lesions), hemorrhages and exudates (bright lesions) and will be according to the labels available in the dataset.

4.6.5 Feature Classification

The extracted features are in turn used in decision making. The decision (DR/Non-DR) will be displayed as an output in the project report.

4.6.6 GUI presentation

The visual demonstration of the project is done through the aid of GUI (Graphical User Interface) using React.

4.7 ASSUMPTIONS AND DEPENDENCIES

Since Google Colaboratory has been used, all the required dependencies like GPU, TensorFlow, Keras library, Numpy, Pandas are installed. The full working of RetinaDoc will depend on the network. So, the status of the internet connection must always be maintained. Ophthalmologists may need to purchase a 20D lens to use a mobile phone to get scans as close as possible to actual retinal scans taken in the hospital.

It is also assumed that the dataset obtained is 100% accurately labelled and the information in the dataset is correct.

Chapter 5

Detailed Design & Architecture

Chapter 5

DETAILED DESIGN AND ARCHITECTURE

5.1 SYSTEM ARCHITECTURE

This section will provide a detailed picture of the projects' architecture including high level system design and UML diagrams depicting the system processes. This project will follow layered architecture model because there is no server between users and system and no database because we get images directly from the system.

5.1.1 Architectural Design

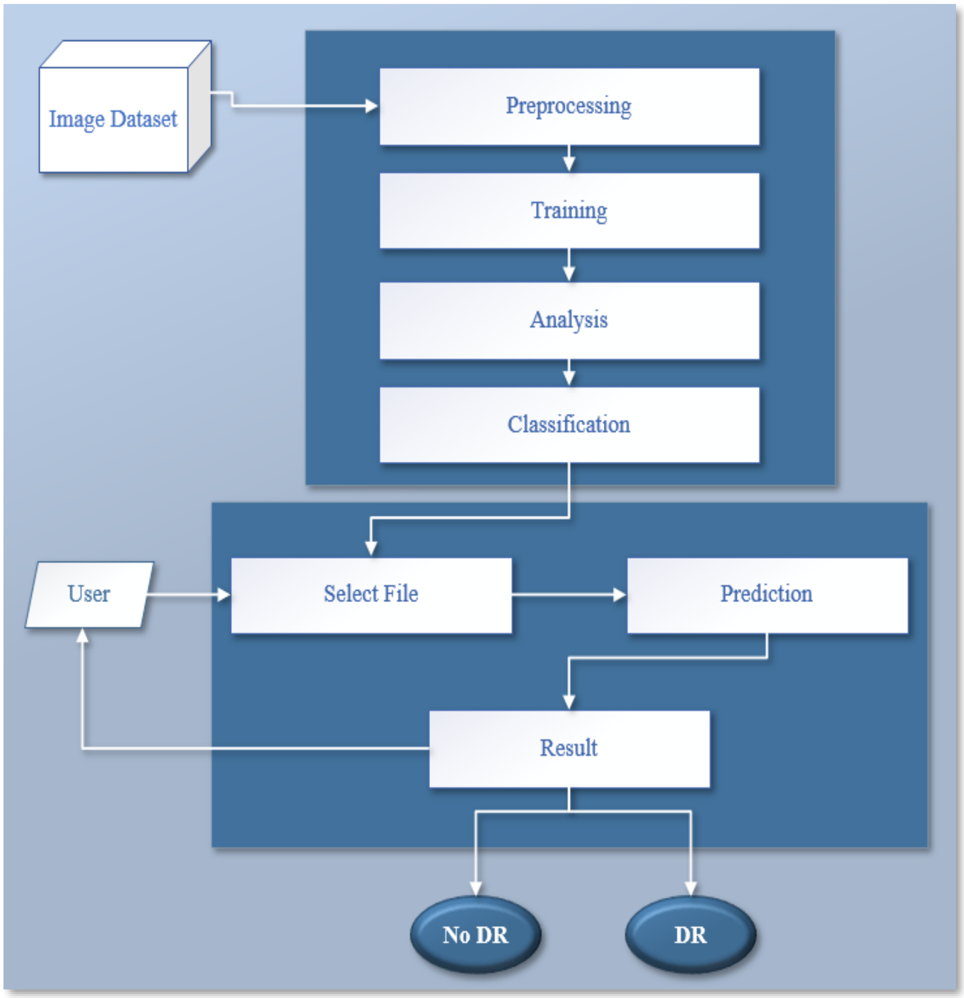


Figure 5. Architectural Diagram

The above image depicts the architecture diagram of RetinaDoc. The image dataset is Preprocessed, in other words data which is gathered from other sources is in raw format and not appropriate for analysis. This data is cleansed and sent to the training model. After analysis of data CNN algorithm is used for classification and then one file is selected by admin to predict the result and the prediction of result is done using tensor-flow Keras library. The result consists of two categories that are No DR and DR.

5.2 DECOMPOSITION DESCRIPTION

The decomposition of the subsystems shown in the architectural design is explained in the following ways.

5.2.1 Module Decomposition

The module decomposition is explained through class diagram that depicts all the classes of the project.

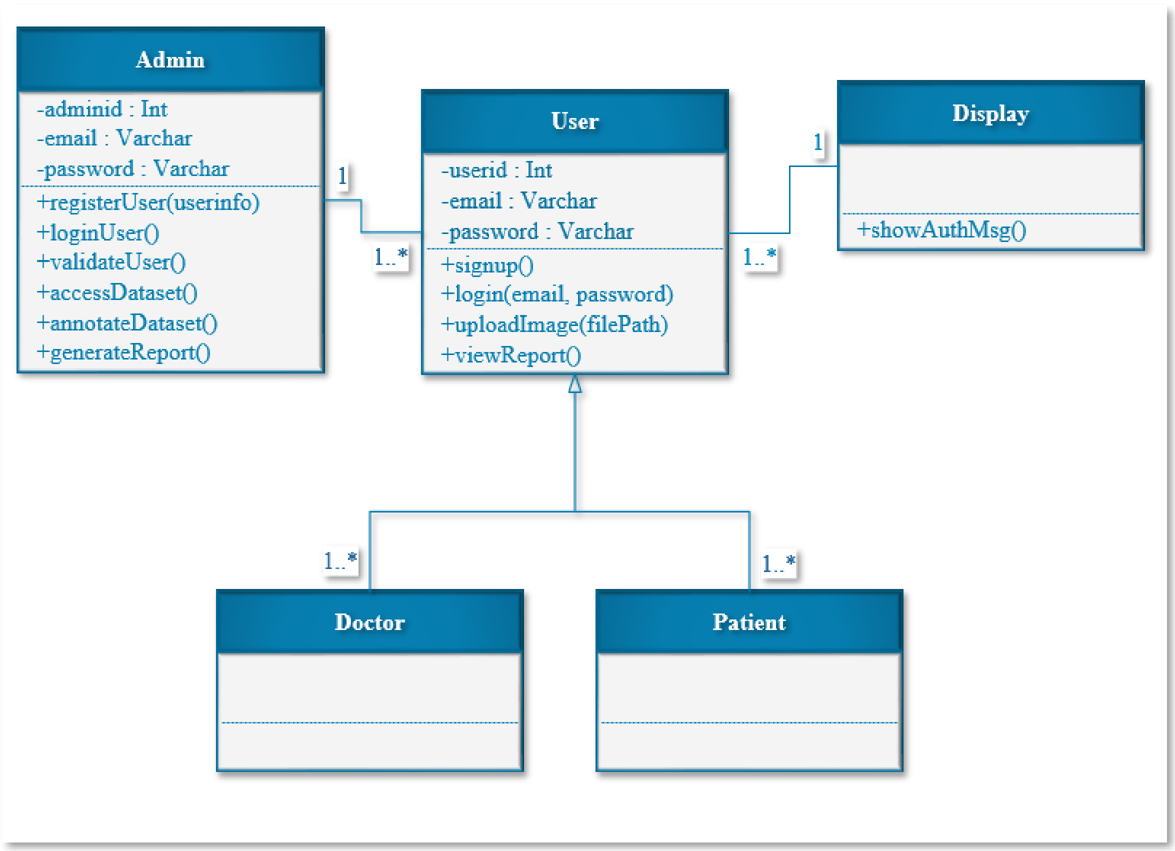


Figure 6. Class Diagram

As shown in the above figure, our project has 4 classes:

1. **Admin Class:** Admin has the attributes of id email and password. The admin works to validate the users and when the user uploads the image admin side accesses the dataset and runs the deep learning algorithm to annotate the dataset, hence, report is generated on the request of user (doctor/patient).

2. **User Class:** User class has attributes of id email and password for the registration and login onto the interface. User class is the parent class of 2 subclasses: patient and doctor who will upload the fundus image to the website to the admin side for the generation of report hence they can view the report

3. **Doctor Class:** Doctor class is the subclass of user class having the attributes and methods inherited by the parent.

4. **Patient Class:** Patient class is the subclass of user class having the attributes and methods inherited by the parent.

5.2.2 Process Decomposition

The process decomposition is explained through use case, sequence and activity diagrams which decompose the system into well-defined and cohesive processes. The following use cases and the subsequent use case narratives explain the set of actions that a user undertakes while dealing with this web-app.

5.2.2.1 Use case diagram

The use cases and the subsequent use case narratives explain the set of actions that an actor undertakes while dealing with RetinaDoc.

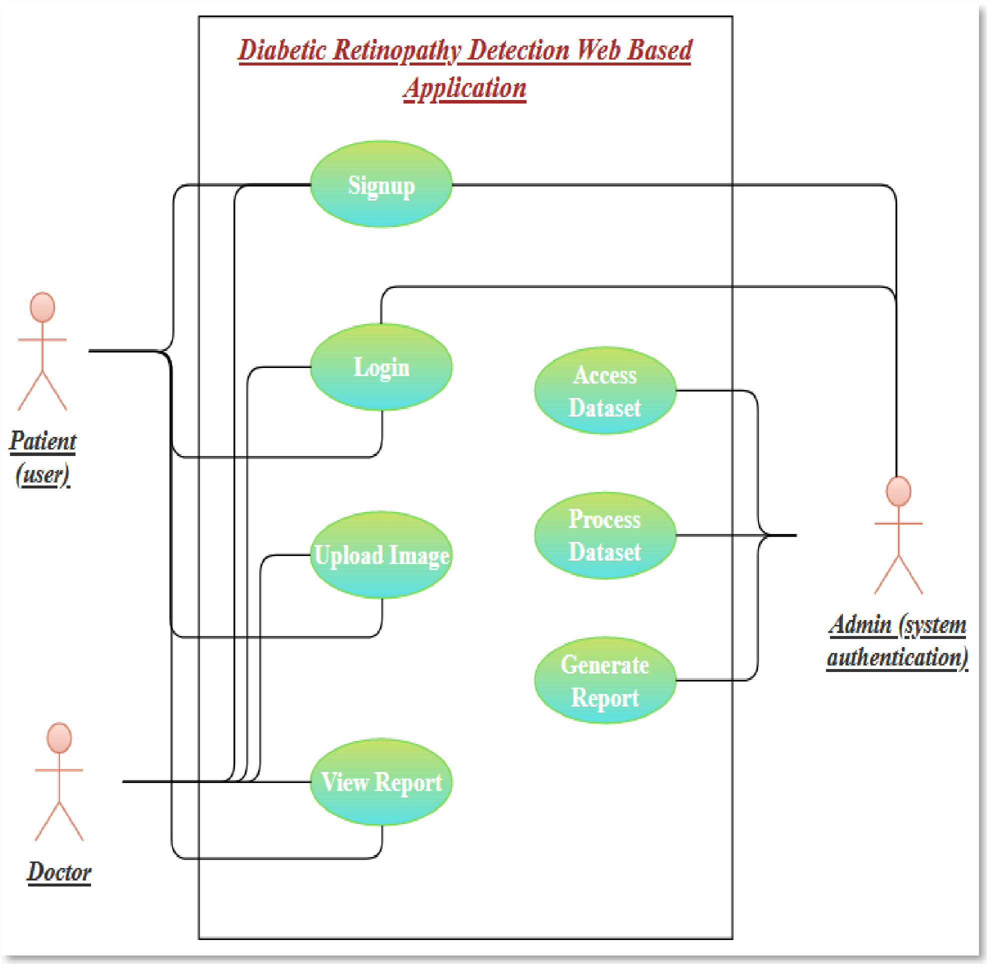


Figure 7. Use case Diagram

5.2.2.2 Use case narratives

1. Patient (user) use cases

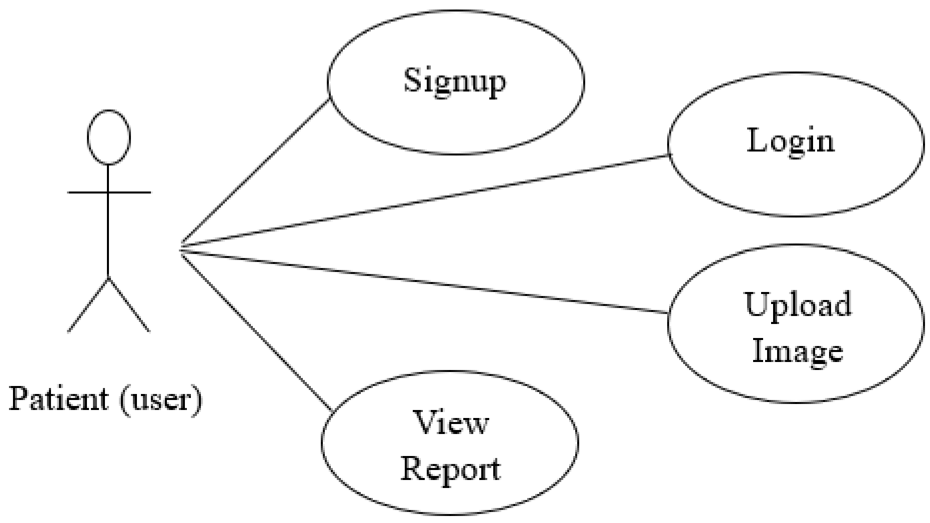
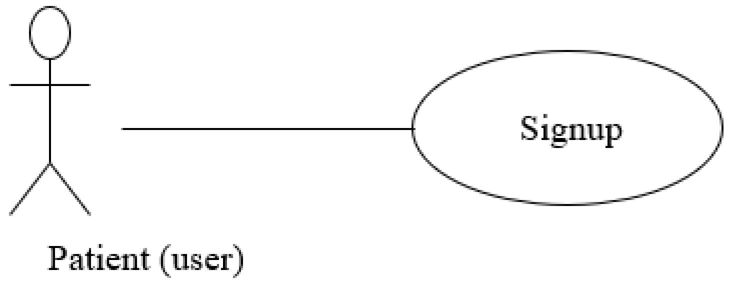


Figure 8. Patient (user) Use Cases

a) ***<u>Use case</u>***: *Signup*

Diagram:



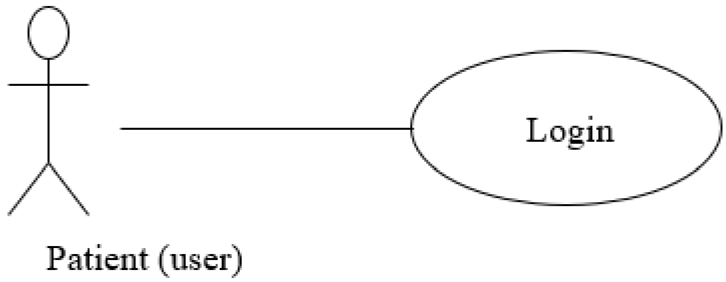
Use case Table:

Table 2. Patient (user) Signup use case narrative

Use Case Name	**Signup**
Precondition	i) A sign-in screen is displayed ii) Must have an internet connection
Basic Path	Once the user signs up, he is directed to the home page of the application.
Postcondition	A confirmation mail is sent to the user of being registered.
Exception Paths	i) If an existing user exists, then a "User already exists" message displayed. ii) If information is missing "Insufficient Information" message is displayed.
Other	None

b) <u>Use case</u>: *Login*

Diagram:



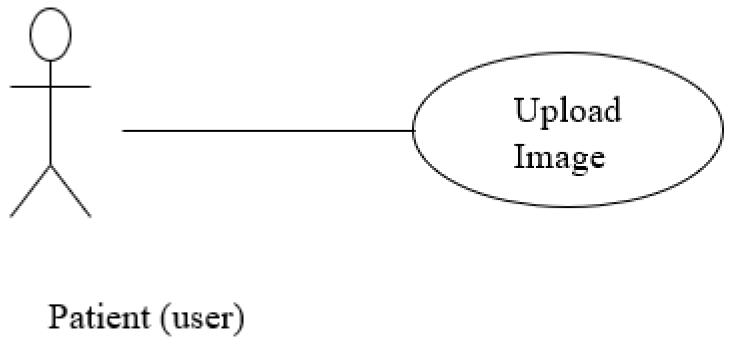
Use case Table:

Table 3. Patient (user) Login Use case Narrative

Use Case Name	**Login**
Precondition	i) The account should already be registered ii) Must have an internet connection iii) Username and Password should be correct
Basic Path	Once the user logs in, he is directed to the home page.
Postcondition	Home screen of the application is displayed.
Exception Paths	i) Alert of Invalid user login details. ii) Requested to re-enter the login details.
Other	None

c) <u>Use case</u>: *Upload Image*

Diagram:



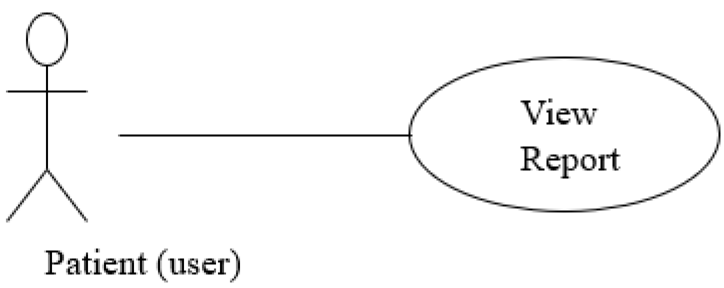
Use case Table:

Table 4. Patient (user) Upload Image use case narrative

Use Case Name	**Upload Image**
Precondition	i) The account should be already logged in. ii) Must have an internet connection.
Basic Path	Once the user clicks the upload image button, user will be allowed to upload the retinal fundus image.
Postcondition	The message is displayed showing image was successfully uploaded.
Exception Paths	The attempt may be abandoned at any time due to poor internet connection or the wrong format of the image.
Other	None

d) <u>Use case</u>: *View Report*

Diagram:



Use case Table:

Table 5. Patient (user) View Report use case narrative

Use Case Name	**View Report**
Precondition	i) Image should be uploaded successfully. ii) A report generated should be sent by the admin. iii) Must have an internet connection.
Basic Path	After successful uploading of image, report generated by admin will be showed.
Postcondition	Report of diabetic retinopathy will be displayed.
Exception Paths	Due to poor connectivity, report unable to be generated hence not received
Other	None

2. Doctor Use Cases

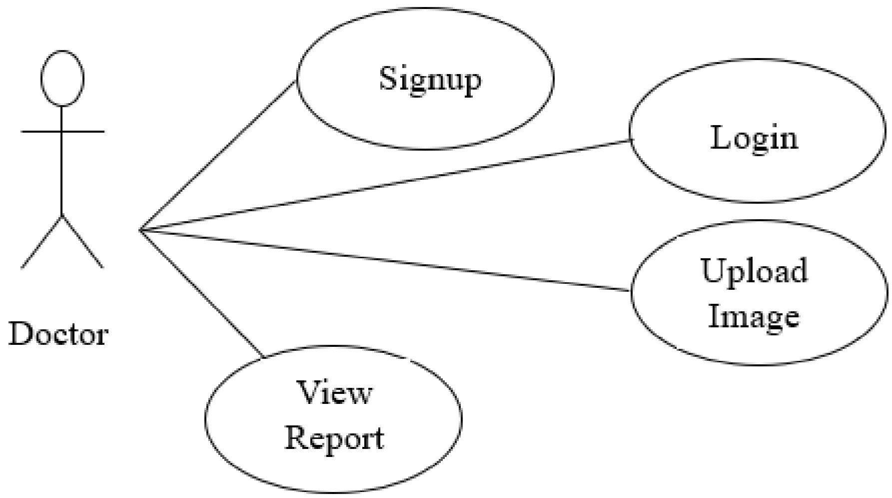
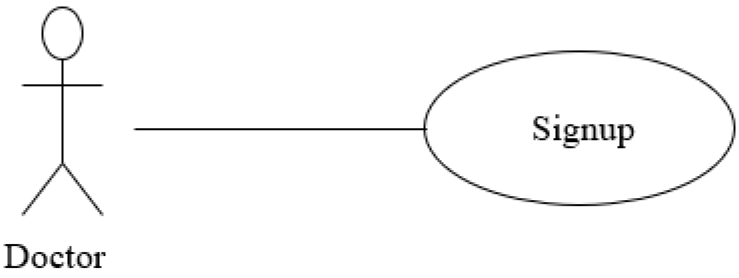


Figure 9. Doctor Use Cases

a) Use case: *Signup*

Diagram:



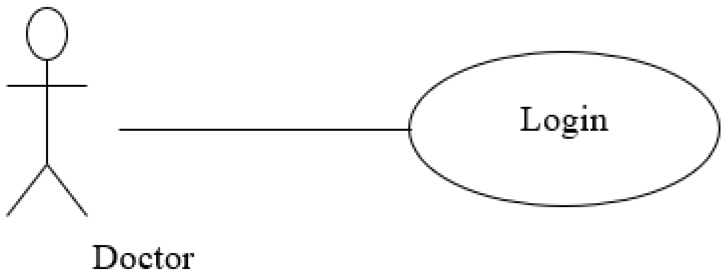
Use case Table:

Table 6. Doctor (user) Signup Use case Narrative

Use Case Name	**Signup**
Precondition	i) A sign-in screen is displayed ii) Must have an internet connection
Basic Path	Once the doctor signs up, he is directed to the home page.
Postcondition	A confirmation registration mail is sent to the user(doctor).
Exception Paths	i) If an existing account exists, then an "Account already exists" message is displayed. ii) If information is missing "Insufficient Information" message is displayed.
Other	None

b) <u>Use case</u>: *Login*

Diagram:



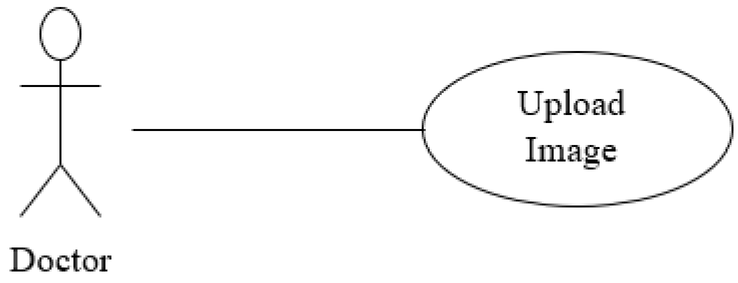
Use case Table:

Table 7. Doctor (user) Login Use case Narrative

Use Case Name	**Login**
Precondition	i) The account should already be registered by the doctor. ii) Must have an internet connection iii) Username and Password should be correct
Basic Path	Once the user(doctor) logs in, he is directed to the home page.
Postcondition	Home screen of the application is displayed.
Exception Paths	i) Invalid user login details ii) Requested to re-enter the login details
Other	None

c) **<u>Use case</u>:** *Upload Image*

Diagram:



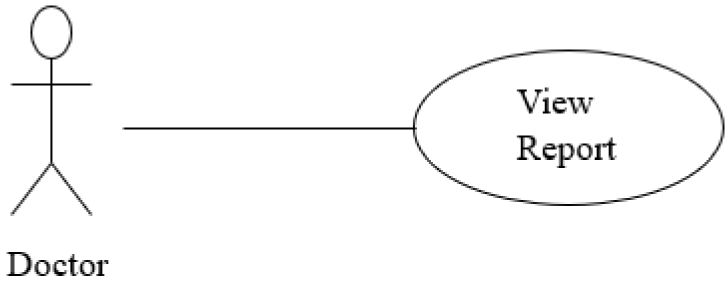
Use case Table:

Table 8. Doctor (user) Upload Image Use case Narrative

Use Case Name	**Upload Image**
Precondition	i) The account should already be logged in. ii) Uploaded picture should be in the right format. iii) Must have an internet connection.
Basic Path	Once the doctor clicks the upload image button, he will be allowed to upload the retinal fundus image.
Postcondition	The message is displayed showing image was successfully uploaded
Exception Paths	The attempt may be abandoned at any time due to weak internet connection or inappropriate format of the uploaded image.
Other	None

d) <u>Use case</u>: *View Report*

Diagram:



Use case Table:

Table 9. Doctor (user) View Report Use case Narrative

Use Case Name	**View Report**
Precondition	i) Image should be uploaded successfully. ii) Reports generated by the admin should be available. iii) Must have an internet connection.
Basic Path	After successful uploading of the image, report generated by admin will be showed.
Postcondition	Report of the diabetic retinopathy will be displayed.
Exception Paths	Due to poor connectivity, report unable to be received.
Other	None

3. **Admin (system authentication) Use Cases**

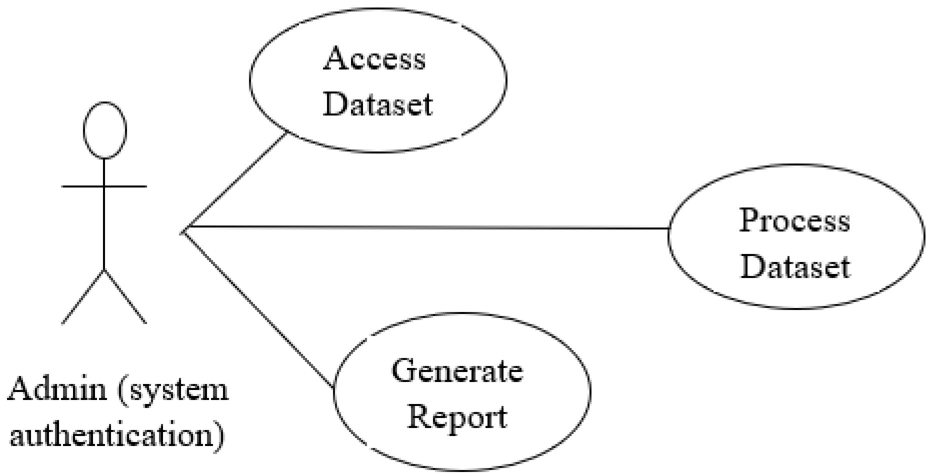
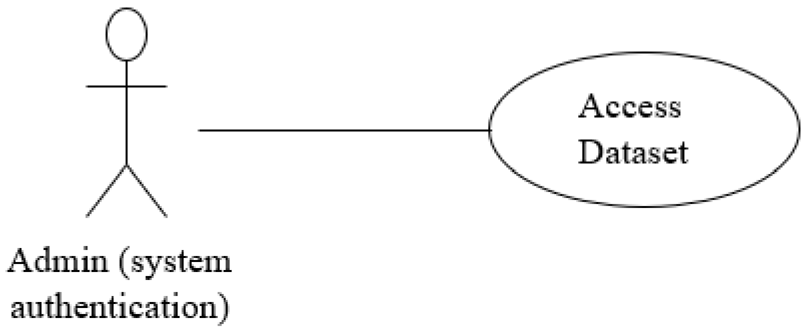


Figure 10. Admin (system authentication) Use Cases

a) **Use case:** *Access Dataset*

Diagram:



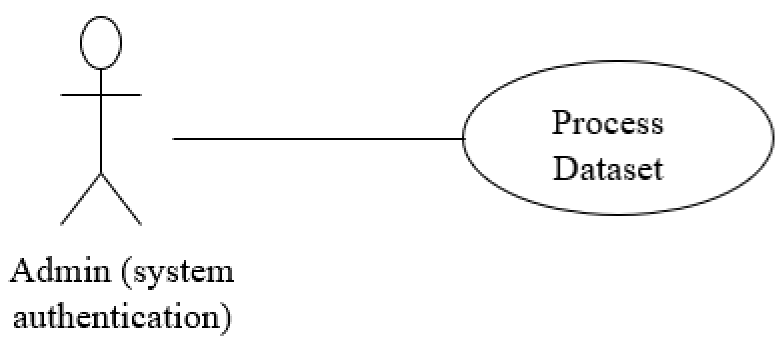
Use case Table:

Table 10. Admin (system authentication) Access Dataset Use case Narrative

Use Case Name	**Access Dataset**
Precondition	i) Image uploading should be successful. ii) Image preprocessing should be done iii) Must have an internet connection.
Basic Path	Admin will be able to access the uploaded retinal images in the dataset.
Postcondition	Machine Learning techniques will be performed on the uploaded retinal images in the dataset.
Exception Paths	i) Garbage image uploaded by user (Patient). ii) Image not uploaded successfully due to poor connection. iii) Image is of poor quality.
Other	None

b) <u>Use case</u>: *Process Dataset*

Diagram:



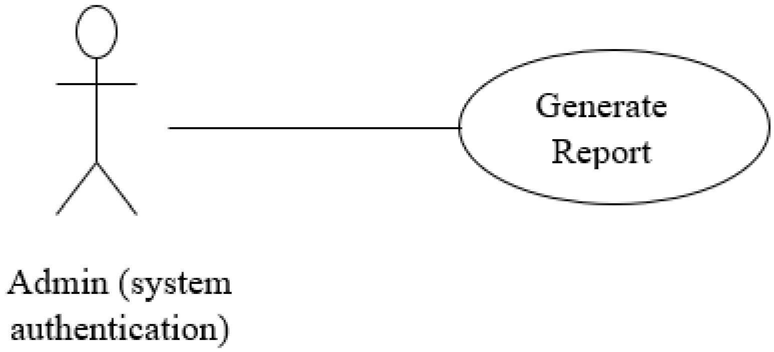
Use case Table:

Table 11. Admin (system authentication) Process Dataset Use case Narrative

Use Case Name	**Process Dataset**
Precondition	i) Image uploading should be successful. ii) Image preprocessing should be done. iii) Dataset should be accessible. iv) Must have an internet connection.
Basic Path	Machine Learning techniques applied on the uploaded images in dataset by clicking on the processing Button.
Postcondition	Results of whether image contains diabetic retinopathy or not will be generated and shown in the form of a report.
Exception Paths	Application stopped due to poor connectivity.
Other	None

c) **Use case**: *Generate Report*

Diagram:



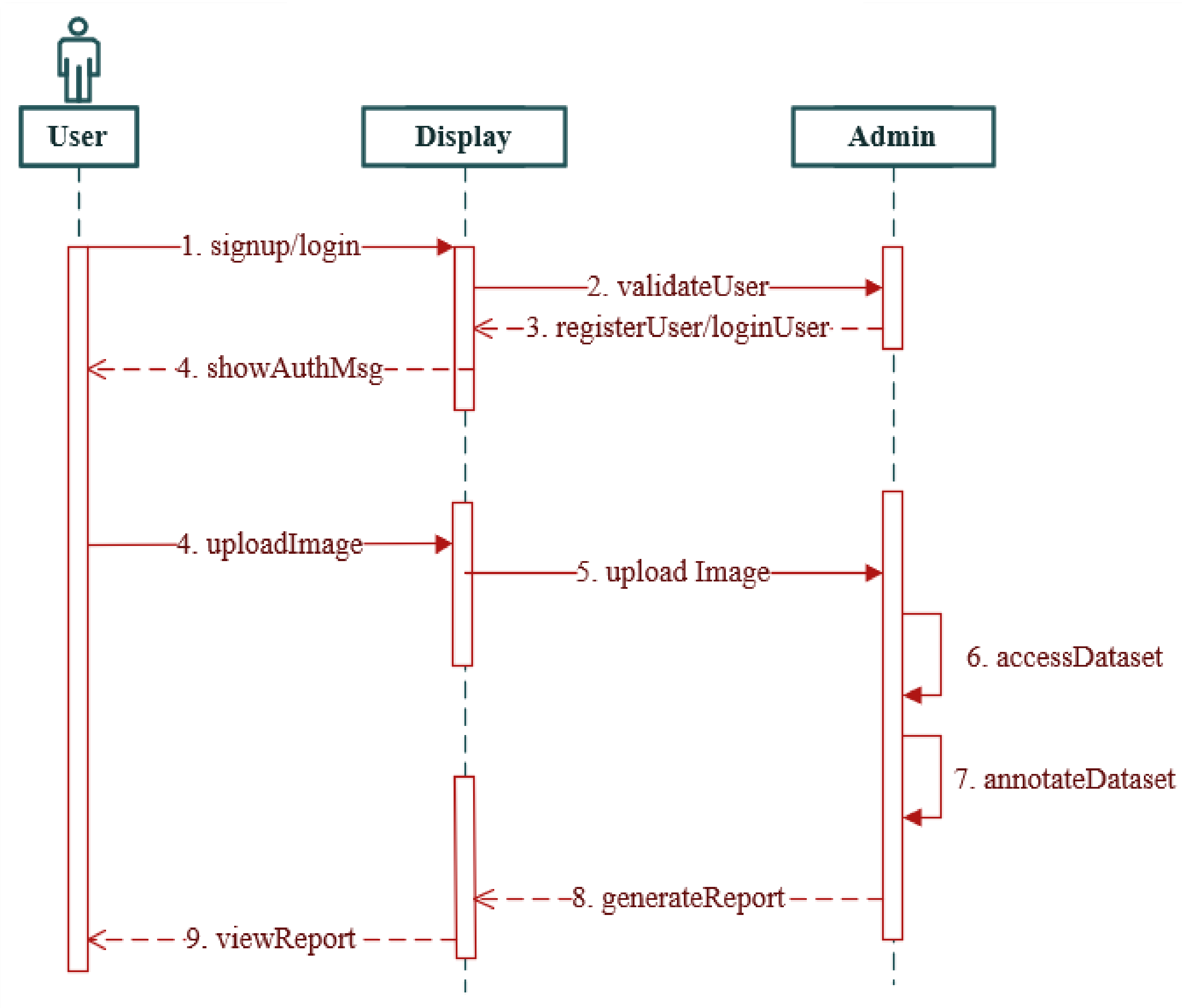
Use case Table:

Table 12. Admin (system authentication) Generate Report Use case Narrative

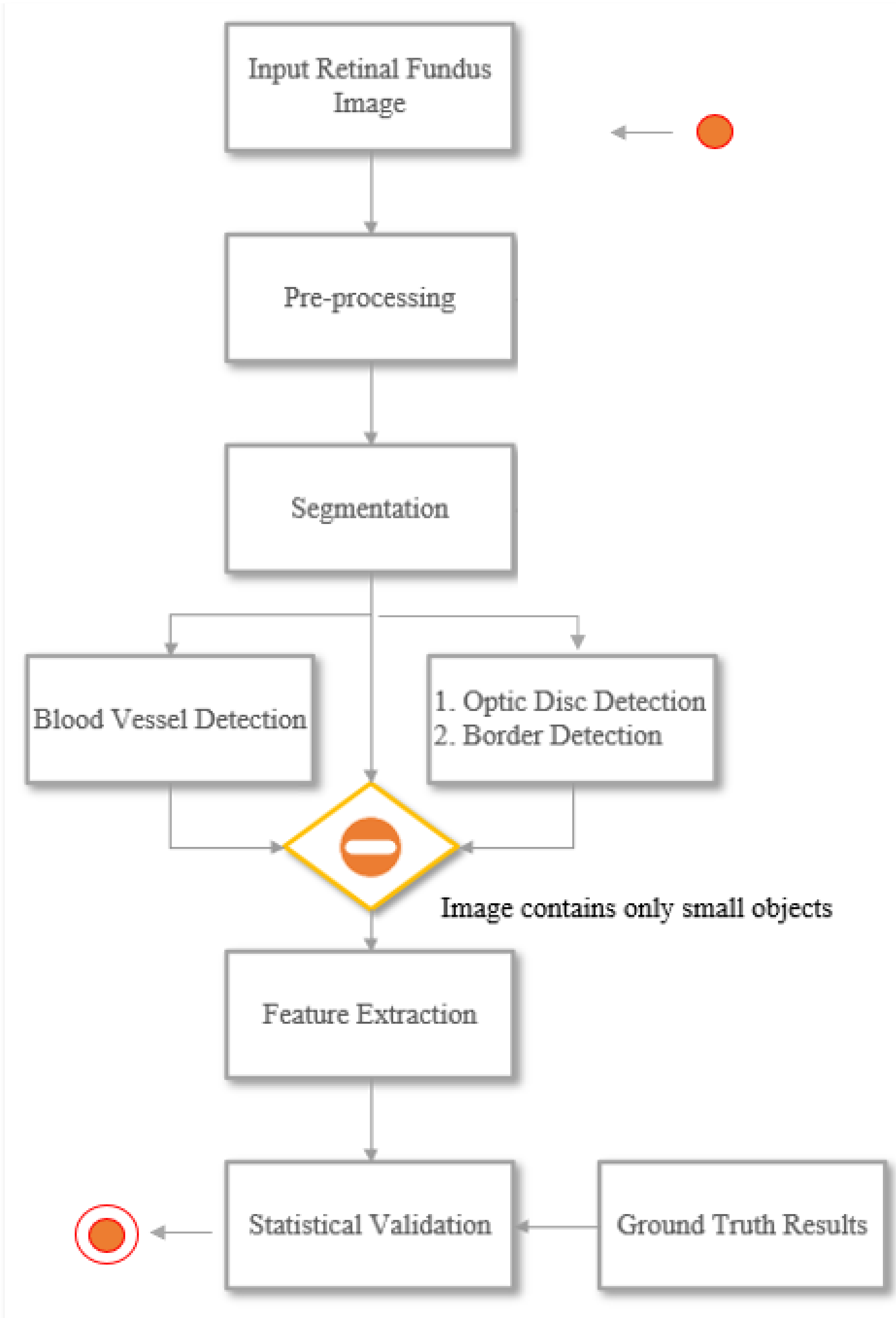
Use Case Name	**Generate Report**
Precondition	i) Processing should be performed on data. ii) Must have an internet connection.
Basic Path	Report of whether the image contains diabetic retinopathy or not, is generated.
Postcondition	The generated report of the patient's result is sent to the user (patient or doctor) whoever requested it and is visible to them.
Exception Paths	Due to poor connectivity, report generation delayed.
Other	None

5.2.2.3 Sequence Diagram

Figure 11. RetinaDoc Sequence Diagram



5.2.2.4 Activity Diagram


Input Retinal Fundus Image
Pre-processing
Segmentation
Blood Vessel Detection
1. Optic Disc Detection
2. Border Detection
Image contains only small objects
Feature Extraction
Statistical Validation
Ground Truth Results

5.3 DESIGN RATIONALE

The design chosen for RetinaDoc is a layered architecture. We've used layered architecture over client server and data storage design as there is no server between users and system and no database because we get images directly from the system.

We will have a 3 layered architecture having Presentation, Application and Data storage Layer.

- **Presentation Layer**: It will handle the view or UI.
- **Application Layer**: It will have all the business logic incorporated.
- **Data storage Layer**: It manages or store data, either locally on client system or on the remote Cloud Server.

Layered architecture is picked up as it helps make small independent components very easily, making proper boundaries around different components, and providing high cohesion.

Clearly, components can do their work independently containing all the required functionality within themselves. That led us to high cohesion. Moreover, components do not have much interaction, once a component has completed its work, it will communicate its state to the other components. Consequently, that component will come into action. That led us to low coupling.

5.4 COMPONENT DESIGN

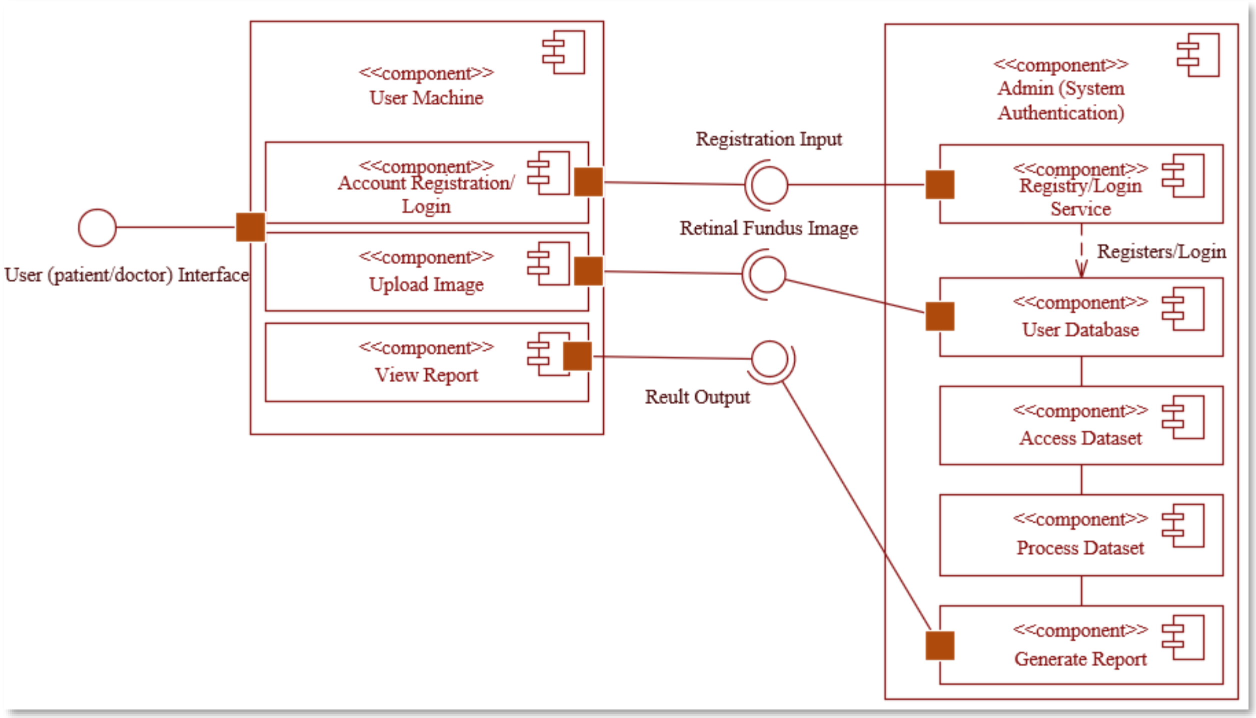


Figure 13. RetinaDoc Component Diagram

5.4.1 User (patient/doctor) Machine

5.4.1.1 Account Registration/Login

Table 13. Account Registration Component Description

Identification	• Name: Account Registration/Login • Location: User Machine Component
Type	Component
Dependencies	No dependencies. Any user with internet connection can register/login to the system.
Function	Users can request for registration by entering valid information, which will be accepted by admin. Then sign in credentials will be provided to the user. Once the user logs in, he is directed to the profile page.
Data	User information

5.4.1.2 Upload Image

Table 14. Upload Image Component Description

Identification	• Name: Upload File • Location: User Machine Component
Type	Component
Dependencies	i) The account should already be registered/logged in. ii) Must have an internet connection. iii) File should be of correct format.
Function	User can upload image files on database
Data	Retinal Fundus Image Files

5.4.1.3 View Report

Table 15. View Report Component Description

Identification	• Name: View Report • Location: User Machine Component
Type	Component
Dependencies	i) Dataset must be processed by Admin. ii) Prior to viewing the report must first be generated by the Admin.
Function	Report can be viewed on user dashboard
Data	Diagnosis Report

5.4.2 Admin (System Authentication)

5.4.2.1 Registry/Login Service

Table 16. Registry/Login Service Component Description

Identification	• Name: View Report • Location: Admin (system authentication) Component
Type	Component.
Dependencies	No dependencies.
Function	Registry/Login Service will respond to user request and take appropriate action.
Data	User information

5.4.2.2 User Database

Table 17.User Database Component Description

Identification	• Name: User Database • Location: Admin (system authentication) Component
Type	Component.
Dependencies	No dependencies.
Function	Registered user will be stored in Database.
Data	User information

5.4.2.3 Access Dataset

Table 18. Access Dataset Component Description

Identification	• Name: Access Dataset • Location: Admin (system authentication) Component
Type	Component.
Dependencies	User must upload the image file before admin can access it.
Function	Admin accesses the user data to process it.
Data	Image File

5.4.2.4 Process Dataset

Table 19. Process Dataset Component Description

Identification	• Name: Process Dataset • Location: Admin (system authentication) Component
Type	Component.
Dependencies	Admin must have access to data file.
Function	Admin accesses the user data to process it.
Data	Processed Image File

5.4.2.5 Generate Report

Table 20. Generate Report Component Description

Identification	• Name: Generate Dataset • Location: Admin (system authentication) Component
Type	Component.
Dependencies	Data file must be processes by the admin before generating report.
Function	Report generation based on processed data file.
Data	Diagnosis Report

Chapter 6

Other Non-Functional Requirements

Chapter 6

OTHER NONFUNCTIONAL REQUIREMENTS

6.1 PERFORMANCE REQUIREMENTS

i) The application will run if the system has reliable internet connection. The application can run on both computers and mobile phones.

ii) The results are generated within minutes and do not take a lot of time thus making response time appropriate and quick for the users.

iii) Response to user errors and undesirable situations have been taken care of to ensure that the application runs without any uncertainty.

iv) Performance of the app also depends on the kind of image of retina scan uploaded by the user; a clearer and proper image ensures better response time.

6.2 SAFETY REQUIREMENTS

RetinaDoc is a consultation app which is not a substitute to any kind of treatment thereby reducing loss, damage and harm resulting from the use of the product. To ensure that the product delivers satisfactory results, the developer team updates the database regularly to avoid incorrect results.

6.3 SECURITY REQUIREMENTS

RetinaDoc does not have any kind of third-party security requirements yet as any type of user can use it without any additional privileges. For the initial stages we are using firewall and network protection of our own windows (domain network, public network, and private network).

For future we plan to broaden our scope with the involvement of third-party network protection.

6.4 SOFTWARE QUALITY ATTRIBUTES

RetinaDoc provides users with both simple and complex features. Due to its well designed and easy to use interface, it can be used by both experts and typical users. However, the users must have a basic knowledge on how to scan and upload an image of the retina scan and interpret the report generated although sufficient help will be provided by the app.

6.5 BUSINESS RULES

For any user to use this application they require to have a retina scan with them. In case the retina scan is unavailable, they need 20D lens to capture the image of retina; although the results will not be as good as the results obtained from a proper retina scan.

Chapter 7

Implementation

Chapter 7

IMPLEMENTATION

7.1 IMPORT LIBRARIES

The code imports various libraries and modules required for building and training deep learning models, as well as for data processing and visualization. It includes specific modules from TensorFlow and Keras, such as ImageDataGenerator, Model, Sequential, pre-trained models, layers, and optimizers. The code also imports utility functions for working with Keras models.

```
from imutils import paths
import matplotlib.pyplot as plt
import argparse
import os
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Model,Sequential
from tensorflow.keras.applications import VGG16, VGG19, InceptionV3, Xception,DenseNet121, MobileNet
from tensorflow.keras.layers import AveragePooling2D, MaxPooling2D,Dropout,Flatten,Dense, Input
from tensorflow.keras.layers import BatchNormalization
from tensorflow.keras.regularizers import l2
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix, roc_auc_score, roc_curve, auc
import numpy as np
import cv2
import os
from matplotlib import ticker
import seaborn as sns
from tensorflow.keras.optimizers import Adam,SGD, Nadam,Ftrl, RMSprop, Adadelta, Adagrad, Adamax
from tensorflow.keras.utils import to_categorical,plot_model
```

Figure 14. Libraries Imported

7.2 IMAGE PREPROCESSING TECHNIQUES

Image preprocessing refers to a set of techniques used to transform raw image data into a format that is better suited for analysis or further processing. This may involve tasks such as resizing, normalization, color conversion, and noise reduction.

The goal of image preprocessing is to enhance the quality and consistency of image data, thereby improving the accuracy and performance of machine learning models or other computer vision algorithms that make use of the data.

The techniques that we've used are as follows:

7.2.1 Resizing and Classification

```
for iPath in iPaths:
    label = iPath.split(os.path.sep)[-2]
    image = cv2.imread(iPath)
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
#   image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    image = cv2.resize(image, (224, 224))
    data.append(image)
    labels.append(label)
```

Figure 15. Code for Resizing and Classification

7.2.2 Data Augmentation

```
#Keras ImageDataGenerator is used to take the inputs
trainAug = ImageDataGenerator(
  rotation_range=15,
  width_shift_range=0.2,
  shear_range=0.2,
  zoom_range=0.2,
  horizontal_flip=True,
  fill_mode="nearest")
```

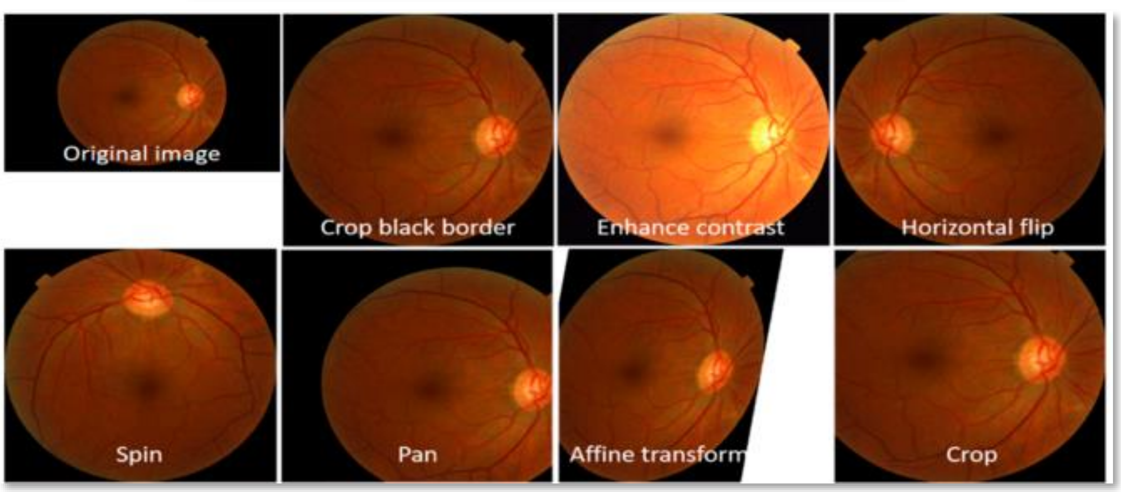


Figure 16. Visual example of the proposed Data Augmentation

7.3 MAKING OF THE MODEL (CNN MODEL)

We have designed a CNN Model to process and analyze images by using multiple layers of convolutional and pooling operations to extract features, followed by fully connected layers for classification. CNNs are particularly effective at capturing spatial relationships and patterns within images and have been used for a wide range of computer vision tasks, such as object detection, segmentation, and recognition.

7.3.1 Xception Model

We have used the Xception model that has advantages over traditional CNNs due to its smaller number of parameters and increased depth and separability of convolutional layers, leading to improved accuracy and computational efficiency in computer vision tasks such as image classification and object recognition

```
bModel = Xception(weights="imagenet", include_top=False, input_tensor=Input(shape=(224, 224, 3)))
```

```
bModel.summary()
```

```
Model: "xception"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_1 (InputLayer)           [(None, 224, 224, 3  0           []
                                )]

 block1_conv1 (Conv2D)          (None, 111, 111, 32  864         ['input_1[0][0]']
                                )

 block1_conv1_bn (BatchNormaliz  (None, 111, 111, 32  128        ['block1_conv1[0][0]']
 ation)                         )

 block1_conv1_act (Activation)  (None, 111, 111, 32  0           ['block1_conv1_bn[0][0]']
                                )

 block1_conv2 (Conv2D)          (None, 109, 109, 64  18432       ['block1_conv1_act[0][0]']
                                )

 block1_conv2_bn (BatchNormaliz  (None, 109, 109, 64  256        ['block1_conv2[0][0]']
 ation)                         )

 block1_conv2_act (Activation)  (None, 109, 109, 64  0           ['block1_conv2_bn[0][0]']
                                )
```

Figure 17. Xception Model

7.3.2 Additional Layers Added (Residual Blocks)

We are not using the Xception Model directly in a sequential manner rather we have added the additional layers (residual blocks → average pooling layer, flatten layer, and dense layers) using skip connection after "block4_sepconv2" creating our own functional model.

```
Last_layer=bModel.get_layer('block4_sepconv2')

print(Last_layer.output_shape)
(None, 28, 28, 728)

Last_output=Last_layer.output

hModel = AveragePooling2D(pool_size=(3, 3))(Last_output) #Average pooling involves

hModel = Flatten(name="flatten")(hModel) #Flattening is used to convert all the res

hModel = Dense(64, activation="relu")(hModel) #Dense Layer is simple layer of neuro

#Each neuron in a neural network computes an output value by applying a specific fu

lhMode = Dense(2, kernel_regularizer='l2', activation ='linear')(hModel)

CAD_HR = Model(inputs=bModel.input, outputs=hModel)
```

Figure 18. Residual Block

7.3.3 Functional Model

The Functional Model that we've created.

```
CAD_HR.summary()

Model: "model"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_1 (InputLayer)           [(None, 224, 224, 3  0           []
                                )]

 block1_conv1 (Conv2D)          (None, 111, 111, 32  864         ['input_1[0][0]']
                                )

 block1_conv1_bn (BatchNormaliz  (None, 111, 111, 32  128        ['block1_conv1[0][0]']
 ation)                         )

 block1_conv1_act (Activation)  (None, 111, 111, 32  0           ['block1_conv1_bn[0][0]']
                                )

 block1_conv2 (Conv2D)          (None, 109, 109, 64  18432       ['block1_conv1_act[0][0]']
                                )

 block1_conv2_bn (BatchNormaliz  (None, 109, 109, 64  256        ['block1_conv2[0][0]']
 ation)                         )

 block1_conv2_act (Activation)  (None, 109, 109, 64  0           ['block1_conv2_bn[0][0]']
                                )

 block2_sepconv1 (SeparableConv  (None, 109, 109, 12  8768       ['block1_conv2_act[0][0]']
 2D)                            8)

 block2_sepconv1_bn (BatchNorma  (None, 109, 109, 12  512        ['block2_sepconv1[0][0]']
 lization)                      8)

 block2_sepconv2_act (Activatio  (None, 109, 109, 12  0          ['block2_sepconv1_bn[0][0]']
 n)                             8)

 block2_sepconv2 (SeparableConv  (None, 109, 109, 12  17536      ['block2_sepconv2_act[0][0]']
 2D)                            8)

 block2_sepconv2_bn (BatchNorma  (None, 109, 109, 12  512        ['block2_sepconv2[0][0]']
 lization)                      8)

 conv2d (Conv2D)                (None, 55, 55, 128)  8192        ['block1_conv2_act[0][0]']

 block2_pool (MaxPooling2D)     (None, 55, 55, 128)  0           ['block2_sepconv2_bn[0][0]']

 batch_normalization (BatchNorm  (None, 55, 55, 128)  512        ['conv2d[0][0]']
 alization)

 add (Add)                      (None, 55, 55, 128)  0           ['block2_pool[0][0]',
                                                                  'batch_normalization[0][0]']

 block3_sepconv1_act (Activatio  (None, 55, 55, 128)  0          ['add[0][0]']
 n)
```

```
 block3_sepconv1 (SeparableConv  (None, 55, 55, 256)  33920      ['block3_sepconv1_act[0][0]']
 2D)

 block3_sepconv1_bn (BatchNorma  (None, 55, 55, 256)  1024       ['block3_sepconv1[0][0]']
 lization)

 block3_sepconv2_act (Activatio  (None, 55, 55, 256)  0          ['block3_sepconv1_bn[0][0]']
 n)

 block3_sepconv2 (SeparableConv  (None, 55, 55, 256)  67840      ['block3_sepconv2_act[0][0]']
 2D)

 block3_sepconv2_bn (BatchNorma  (None, 55, 55, 256)  1024       ['block3_sepconv2[0][0]']
 lization)

 conv2d_1 (Conv2D)              (None, 28, 28, 256)  32768      ['add[0][0]']

 block3_pool (MaxPooling2D)     (None, 28, 28, 256)  0          ['block3_sepconv2_bn[0][0]']

 batch_normalization_1 (BatchNo  (None, 28, 28, 256)  1024       ['conv2d_1[0][0]']
 rmalization)

 add_1 (Add)                    (None, 28, 28, 256)  0          ['block3_pool[0][0]',
                                                                  'batch_normalization_1[0][0]']

 block4_sepconv1_act (Activatio  (None, 28, 28, 256)  0          ['add_1[0][0]']
 n)

 block4_sepconv1 (SeparableConv  (None, 28, 28, 728)  188672     ['block4_sepconv1_act[0][0]']
 2D)

 block4_sepconv1_bn (BatchNorma  (None, 28, 28, 728)  2912       ['block4_sepconv1[0][0]']
 lization)

 block4_sepconv2_act (Activatio  (None, 28, 28, 728)  0          ['block4_sepconv1_bn[0][0]']
 n)

 block4_sepconv2 (SeparableConv  (None, 28, 28, 728)  536536     ['block4_sepconv2_act[0][0]']
 2D)

 average_pooling2d (AveragePool  (None, 9, 9, 728)   0          ['block4_sepconv2[0][0]']
 ing2D)

 flatten (Flatten)              (None, 58968)        0          ['average_pooling2d[0][0]']

 dense (Dense)                  (None, 64)           3774016    ['flatten[0][0]']

 dense_1 (Dense)                (None, 2)            130        ['dense[0][0]']

==================================================================================================
Total params: 4,695,578
Trainable params: 3,774,146
Non-trainable params: 921,432
__________________________________________________________________________________________________
```

Figure 19. Functional Model

7.4 TRAINING THE CNN MODEL

The code block is training a deep learning model called CAD_HR using a generator that augments the training data in real-time. It sets the number of steps per epoch and validation steps based on the batch size and length of training and testing data, respectively. The model is trained for a specified number of epochs, which is also defined in the code block.

```
R = CAD_HR.fit_generator(
    trainAug.flow(X_train, Y_train, batch_size=BS),
    steps_per_epoch=len(X_train) // BS,
    validation_data=(X_test, Y_test),
    validation_steps=len(X_test) // BS,
    epochs=EPOCHS)
```

Figure 20. Code for Training the Model

```
Epoch 1/20
37/37 [==============================] - 206s 5s/step - loss: 8.0649 - accuracy: 0.7243 - val_loss: 0.5831 - val_accuracy:
0.8775
Epoch 2/20
37/37 [==============================] - 179s 5s/step - loss: 0.1822 - accuracy: 0.9632 - val_loss: 0.0253 - val_accuracy:
1.0000
Epoch 3/20
37/37 [==============================] - 139s 4s/step - loss: 0.0967 - accuracy: 0.9812 - val_loss: 0.0394 - val_accuracy:
1.0000
Epoch 4/20
37/37 [==============================] - 119s 3s/step - loss: 0.0534 - accuracy: 0.9923 - val_loss: 0.0343 - val_accuracy:
1.0000
Epoch 5/20
37/37 [==============================] - 140s 4s/step - loss: 0.0361 - accuracy: 0.9974 - val_loss: 0.0205 - val_accuracy:
1.0000
Epoch 6/20
37/37 [==============================] - 123s 3s/step - loss: 0.0431 - accuracy: 0.9923 - val_loss: 0.0196 - val_accuracy:
1.0000
Epoch 7/20
37/37 [==============================] - 112s 3s/step - loss: 0.0237 - accuracy: 1.0000 - val_loss: 0.0353 - val_accuracy:
1.0000
Epoch 8/20
37/37 [==============================] - 112s 3s/step - loss: 0.0254 - accuracy: 0.9991 - val_loss: 0.0182 - val_accuracy:
1.0000
Epoch 9/20
37/37 [==============================] - 115s 3s/step - loss: 0.0357 - accuracy: 0.9940 - val_loss: 0.0177 - val_accuracy:
1.0000
Epoch 10/20
37/37 [==============================] - 112s 3s/step - loss: 0.0398 - accuracy: 0.9983 - val_loss: 0.0173 - val_accuracy:
1.0000
Epoch 11/20
37/37 [==============================] - 112s 3s/step - loss: 0.1680 - accuracy: 0.9846 - val_loss: 0.0662 - val_accuracy:
1.0000
Epoch 12/20
37/37 [==============================] - 113s 3s/step - loss: 0.0608 - accuracy: 0.9957 - val_loss: 0.0166 - val_accuracy:
1.0000
Epoch 13/20
37/37 [==============================] - 113s 3s/step - loss: 0.0253 - accuracy: 0.9991 - val_loss: 0.0163 - val_accuracy:
1.0000
Epoch 14/20
37/37 [==============================] - 112s 3s/step - loss: 0.0380 - accuracy: 0.9957 - val_loss: 0.0160 - val_accuracy:
1.0000
Epoch 15/20
37/37 [==============================] - 113s 3s/step - loss: 0.0253 - accuracy: 0.9991 - val_loss: 0.0301 - val_accuracy:
1.0000
Epoch 16/20
37/37 [==============================] - 113s 3s/step - loss: 0.0192 - accuracy: 1.0000 - val_loss: 0.0154 - val_accuracy:
1.0000
Epoch 17/20
37/37 [==============================] - 114s 3s/step - loss: 0.0217 - accuracy: 0.9991 - val_loss: 0.0152 - val_accuracy:
1.0000
Epoch 18/20
37/37 [==============================] - 112s 3s/step - loss: 0.0302 - accuracy: 0.9983 - val_loss: 0.0149 - val_accuracy:
1.0000
Epoch 19/20
37/37 [==============================] - 114s 3s/step - loss: 0.6290 - accuracy: 0.9486 - val_loss: 0.1891 - val_accuracy:
0.9425
Epoch 20/20
37/37 [==============================] - 113s 3s/step - loss: 0.0526 - accuracy: 0.9966 - val_loss: 0.0144 - val_accuracy:
1.0000
```

7.5 EVALUATION TECHNIQUES

7.5.1 Accuracy and loss

```
plt.xlabel('Epochs')
plt.ylabel('Accuracy and Loss')
plt.plot(R.history['loss'], label='Train loss')
plt.plot(R.history['val_loss'], label='Validation loss')
plt.plot(R.history['accuracy'], label='Train acc')
plt.plot(R.history['val_accuracy'], label='Validation acc')
plt.legend(loc="right")
plt.ylim(0,1.02)
plt.xlim(0,10)
plt.show()
```

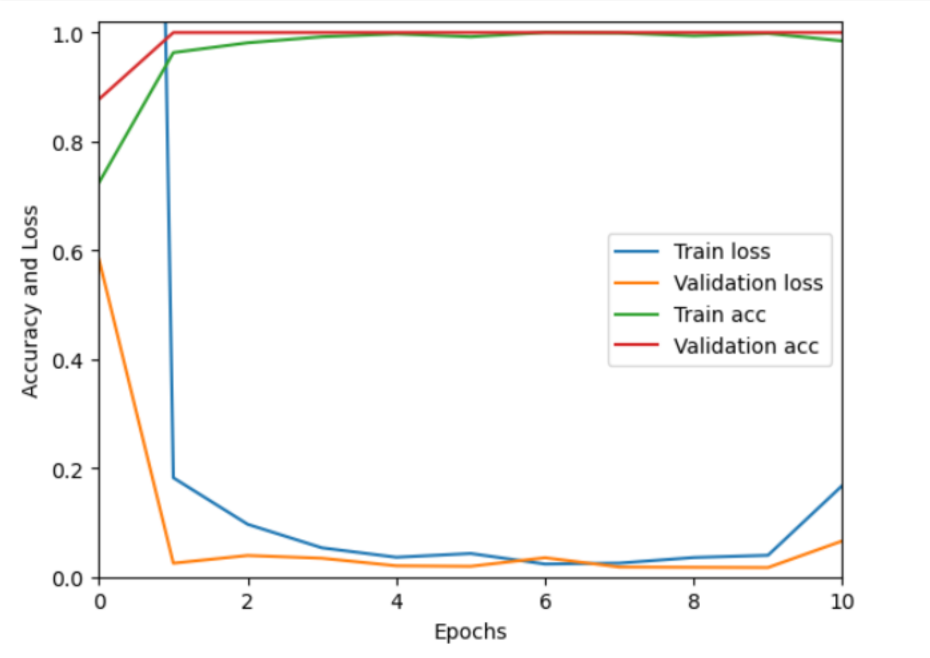


Figure 21. Accuracy and Loss Calculated

7.5.2 Precision, Recall, and f1 Score

```
CAD_HR_pred = np.argmax(CAD_HR_pred, axis=1)
print(classification_report(Y_test.argmax(axis=1), CAD_HR_pred,
    target_names=LB.classes_))
```

```
              precision    recall  f1-score

     Disease       1.00      1.00      1.00
      Normal       1.00      1.00      1.00

    accuracy                           0.97
   macro avg       1.00      1.00      1.00
weighted avg       1.00      1.00      1.00
```

Figure 22. Precision, Recall, and f1 Score Calculated

7.5.3 Accuracy, Sensitivity, and Specificity

```
cm = confusion_matrix(Y_test.argmax(axis=1), CAD_HR_pred)
```

```
total = sum(sum(cm))
```

```
acc = (cm[0, 0] + cm[1, 1]) / total
```

```
sensitivity = cm[0, 0] / (cm[0, 0] + cm[0, 1])
```

```
specificity = cm[1, 1] / (cm[1, 0] + cm[1, 1])
```

```
print("acc: {:.4f}".format(acc))
print("sensitivity: {:.4f}".format(sensitivity))
print("specificity: {:.4f}".format(specificity))
```

```
acc:  0.97
sensitivity: 1.0000
specificity: 1.0000
```

Figure 23. Accuracy, Sensitivity, and Specificity Calculated

7.5.4 Confusion Matrix

```
sns.heatmap(cm, annot=True, fmt='g')
```

```
&lt;AxesSubplot:&gt;
```

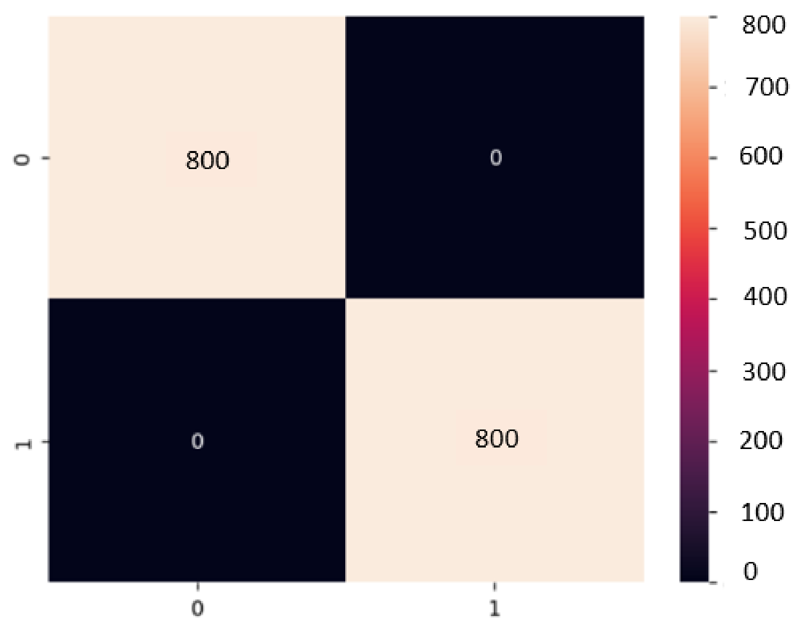


Figure 24. Confusion Matrix Generated

7.6 PREDICTION

```
answers = ["Disease", "Normal"]
for i in range(10):
    plt.imshow(X_test[i])
    plt.title("Predicted : "+ str(answers[predslist[i]]))
    plt.show()
```

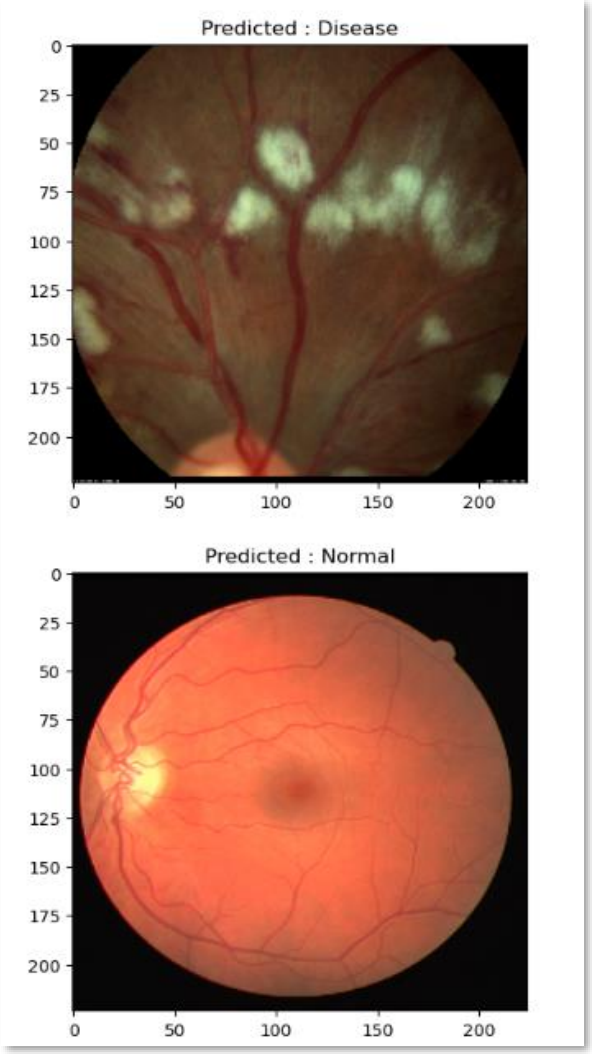


Figure 25. Prediction (Diseased/Non-Diseased)

Finally, the model successfully predicts the binary classification of Disease or No-Disease.

Chapter 8

Testing, Results, and Discussion

Chapter 8

TESTING, RESULTS AND DISCUSSION

The relevant strategies, process and techniques utilized to design, execute, and manage testing of the "Diabetic Retinopathy Detection Web-Based Application (Retina Doc)" are described in this chapter of test plan. The test strategy will verify that the application's criteria and standards are met to an approved level by the end user.

Manual testing will be used, which entails testing software without the use of any automated tools or scripts. In this scenario, the tester assumes the role of an end-user and tests the application for any abnormal behavior or bugs. All functional, application performance and use case criteria mentioned in the requirement document are covered by the test scope. Software testing can be done at any point during the development phase, depending on the testing approach used. However, after the requirements have been developed and the coding process has been done, most of the testing work happens.

8.1 TEST ITEMS

- Development of test scenarios
- Execute multiple tests based on the above-mentioned test scenarios that have been generated
- Inform the appropriate developer or management about any bugs
- Develop and provide test results
- Manage or incorporate adjustments at a later stage in projects development

8.2 FEATURES REQUIRED TO BE TESTED

Following features are used:

- User will be able to open the web page by entering URL
- User will be able to proceed to Retina Docs' main page
- User will be able to sign up for an account
- Existing user will be able to log in to their account
- User will be able to upload his/her retinal scan
- User will be able to view the report generated

Unit Testing: Unit testing is the responsibility of developers. Each module's component's implementation will be checked individually.

Integration Testing: The integration test case will be executed when the unit test has passed over the chosen quality level. It's critical to test the product as a black-box after all the modules have been integrated.

Positive and Negative Testing: This method will be used in conjunction with unit and integration testing. Test cases are written in scenarios that are evident and guarantee that all functional criteria are met. Furthermore, many test cases will be presented to demonstrate how the program responds to invalid operations.

Functional Testing: Is a software testing technique that evaluates the application's functionality by verifying that it meets the specified requirements. This type of testing checks the features and functions of the software, such as input/output processing, data manipulation, user interface, and error handling. It can be performed manually or with the help of automated tools, and it can be executed at various stages of the software development lifecycle. The goal of functional testing is to ensure that the software application functions as intended and meets the needs of the end-users.

Whitebox Testing: Is a software testing technique that evaluates the internal workings or structure of an application. This testing method is also known as clear-box testing, structural testing, or code-based testing. White-box testing involves examining the application's code to ensure that it meets the expected requirements and specifications. Testers performing white-box testing have access to the application's source code and design documents and use this knowledge to create test cases that ensure all the lines of code, branches, and loops are tested. The purpose of white-box testing is to identify coding errors, logic errors, and other defects in the software's internal workings, which can help improve the quality and reliability of the application.

Blackbox Testing: Is a software testing technique that evaluates the functionality of an application without examining its internal structure or workings. This testing method involves testing the software from the perspective of an end-user, without any knowledge of the software's implementation details. Testers performing black-box testing focus on the software's external behavior and use various inputs to verify the expected outputs, without knowing how the application produces the output. Black-box testing helps to identify errors or defects in the software's functionality, as well as the issues in the user interface and integration with other systems

Performance Testing: Is a type of software testing that evaluates the performance of an application under a specific workload. The goal of performance testing is to determine the application's speed, scalability, and stability under various conditions such as high traffic, heavy load, and stress. This type of testing can be done using different techniques such as load testing, stress testing, endurance testing, and spike testing.

8.3 PASS OR FAIL MEASURE

Details of the test cases are specified in section Test Deliverables. Following the principles written below, a test item would be judged as pass or fail.

- Pre-conditions satisfied
- Inputs carried out as per plan

- The output matches what was specified in output → Passed
- The system does not function or does not meet the output requirements → Failed

8.4 STANDARD FOR DEFERRAL AND RENEWAL REQUIREMENTS

Developers can rapidly correct any flaws discovered, eliminating the need to restart the testing process from the start. However, when serious flaws prevent certain test cases from running because they are interdependent, testing must be suspended.

8.5 TEST DELIVERABLES

Following are the Test Cases:

Table 21. Test Case – 01 (Opening the Application)

Test Case Number	**1**
Test Case Name	Open Retina Doc
Description	Testing application whether it runs on web browser or not
Testing Techniques	Unit Testing, Black Box Testing
Preconditions	Web browser will be there on your system
Steps	i) Run server through command prompt ii) Enter web browser (localhost: 8000)
Output	Web Application successfully open and running.
Status	Test Case Passed Successfully

Table 22. Test Case – 02 (Opening the Application - Invalid Case)

Test Case Number	**2**
Test Case Name	Open Retina Doc
Description	Testing application whether it runs on web browser or not
Testing Techniques	Negative Testing
Preconditions	Web browser will be there on your system
Steps	i) Run server through command prompt with limited internet connection ii) Enter web browser through localhost: 8080
Output	Web Application should not open
Status	Test Case Passed Successfully

Table 23. Test Case – 03 (Account Sign up)

Test Case Number	**3**
Test Case Name	Account Sign up
Description	Testing user account is successfully registered or not.
Testing Techniques	Blackbox Testing, Functionality Testing, Unit Testing
Preconditions	Web Application is already opened successfully with sign up page being displayed
Steps	i) Enter Users' first name, last name, email, & password ii) Click the Sign-Up Button
Output	User Account Successfully Registered
Status	Test Case Passed Successfully

Table 24. Test Case – 04 (Account Sign up - Invalid Case)

Test Case Number	**4**
Test Case Name	Account Sign up
Description	Testing user account is successfully registered or not.
Testing Techniques	Negative Testing
Preconditions	Web Application is already opened successfully with sign up page being displayed
Steps	i) Enter User info for an already registered User Account ii) Enter email address in the wrong format iii) Entering Password shorter than 8 characters iv) Click the Signup Button
Output	User Account NOT Registered. Messages Displayed: • Users with this email address already exists. • Enter a Valid email address. • Password must be 8 characters long. • This field is required.
Status	Test Case Passed Successfully

Table 25. Test Case – 05 (Account Sign in)

Test Case Number	**5**
Test Case Name	Account Sign in
Description	Testing user account is successfully logged in or not.
Testing Techniques	Blackbox Testing, Functionality Testing, Unit Testing
Preconditions	Web Application is opened successfully with account already registered; login screen is visible
Steps	i) Enter User credentials ii) Click the Sign in Button
Output	User Account Successfully Logged in
Status	Test Case Passed Successfully

Table 26. Test Case – 06 (Account Sign in - Invalid Case)

Test Case Number	**6**
Test Case Name	Account Sign in
Description	Testing user account is successfully signed in or not.
Testing Techniques	Negative Testing
Preconditions	Web Application is already opened successfully with user account already registered, and sign in page is visible
Steps	i) Enter User Credentials for a non-registered account ii) Click the Sign in Button
Output	User Account NOT Logged in. Messages Displayed: No Active account found with the given credentials
Status	Test Case Passed Successfully

Table 27. Test Case – 07 (Upload Image)

Test Case Number	**7**
Test Case Name	Upload Image
Description	Browse file to upload
Preconditions	The user is already logged in to his/her accounts dashboard.
Testing Technique	Unit Testing, Black Box Testing
Steps	Either drag and drop the required file or click to browse the image.
Expected Output	Image successfully uploaded.
Actual Output	Retinal Fundus Scan successfully uploaded.
Status	Test Case passed successfully

Table 28. Test Case – 08 (Generate Report)

Test Case Number	**8**
Test Case Name	Generate Report
Description	Report is generated upon successfully uploading the retinal fundus images
Precondition	Fundus image is already uploaded
Testing Technique	Unit Testing, Black Block Testing
Steps	Click on Generate Report button
Output	Report will be generated
Status	Test Case successfully passed

8.6 RESULTS AND DISCUSSION

Passing all test cases for our Diabetic Retinopathy web application is a significant milestone that assures the application's reliability and readiness for deployment.

It confirms that the application meets its intended requirements as expected, and is free from any significant defects that could compromise its performance, security, or accuracy in detecting diabetic retinopathy using techniques such as:

- Unit Testing
- Positive and Negative Testing
- Integration Testing
- Functional Testing
- Whitebox/Blackbox Testing

This is crucial for an application that plays a critical role in detecting and preventing blindness among diabetes patients. The successful passing of all test cases instills confidence in both the development team and end-users that the application will function optimally and provide valuable insights for the diagnosis and treatment of diabetic retinopathy.

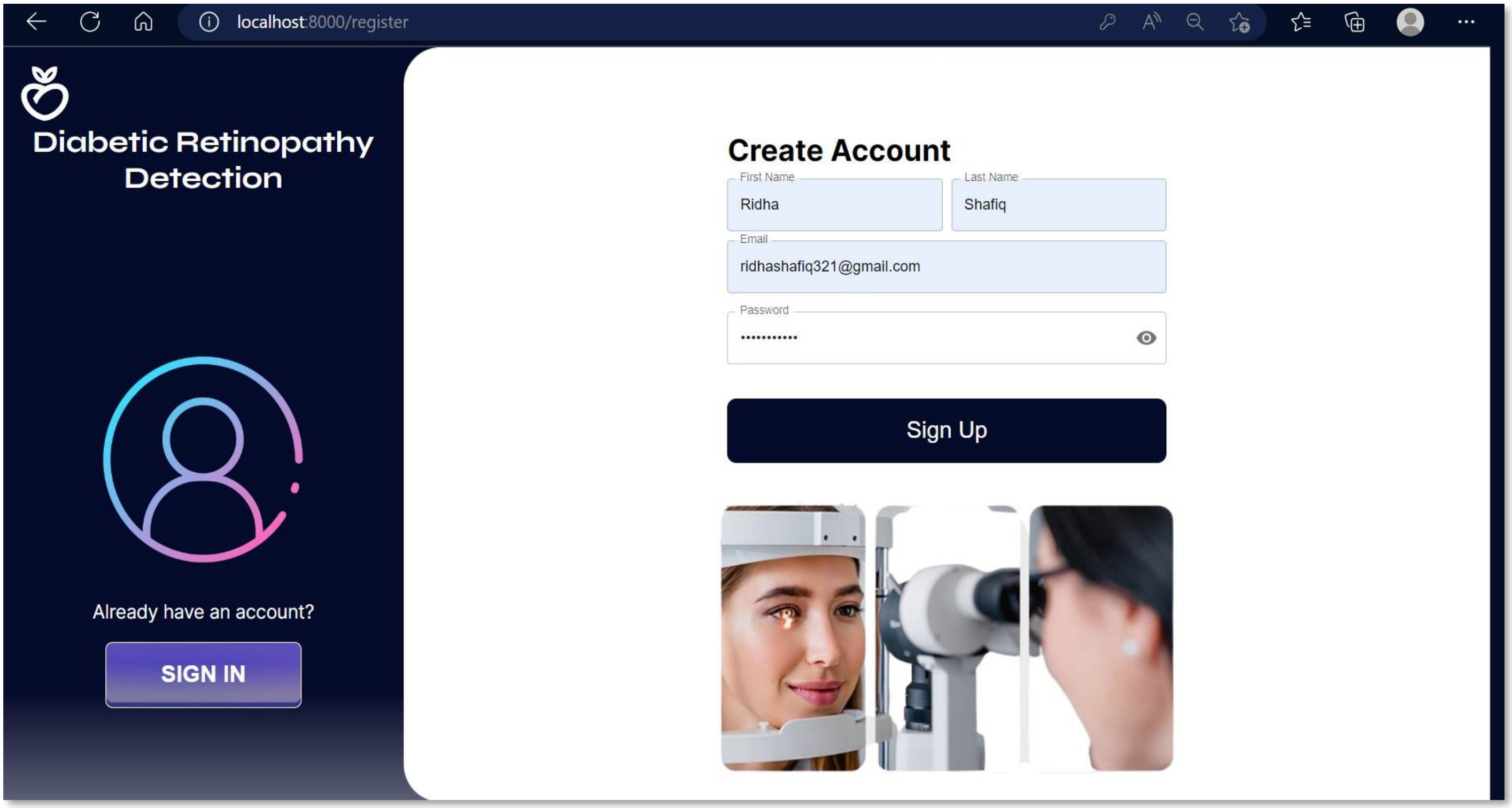
Chapter 9

Interface

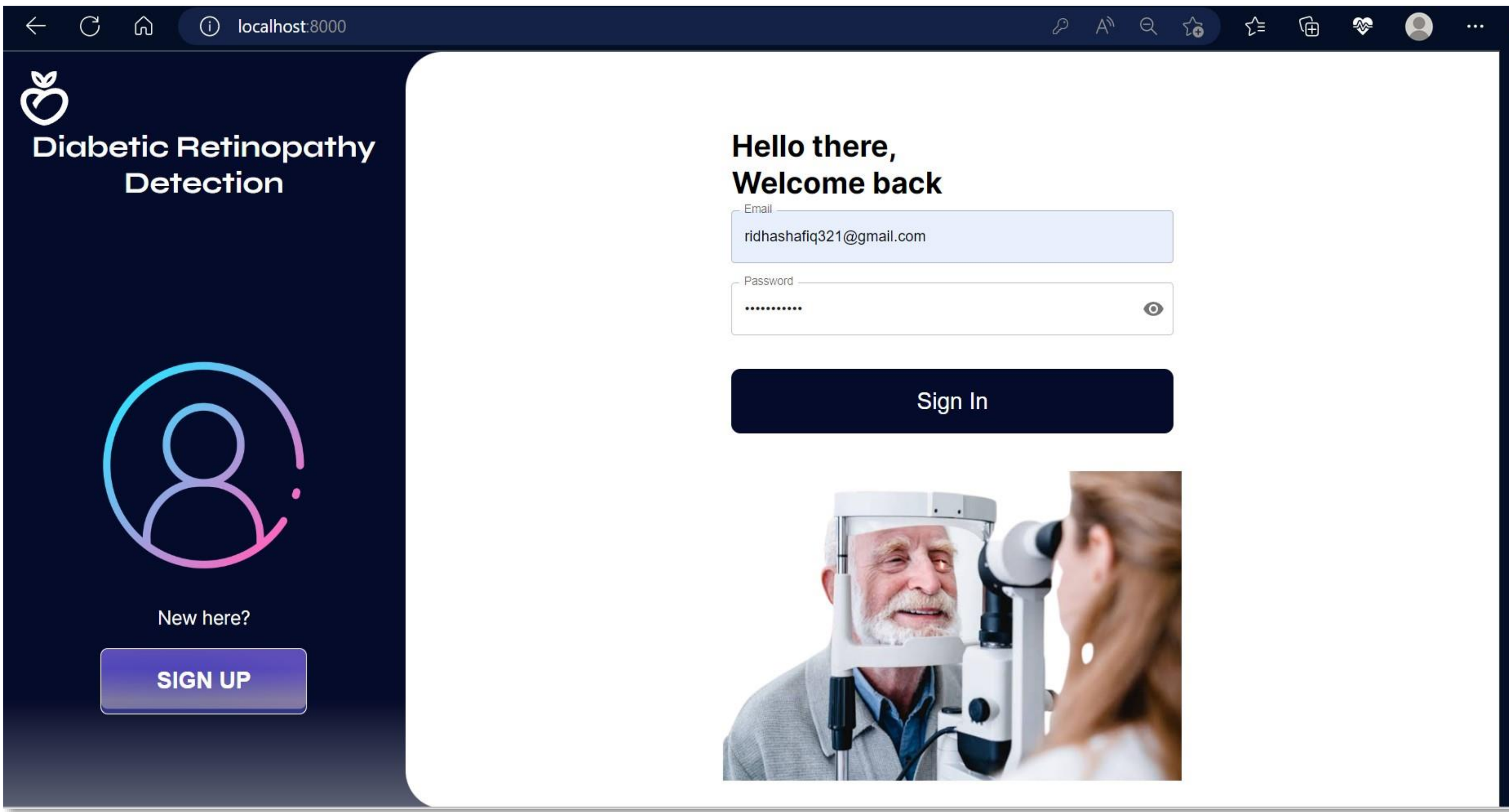
Chapter 9

INTERFACE

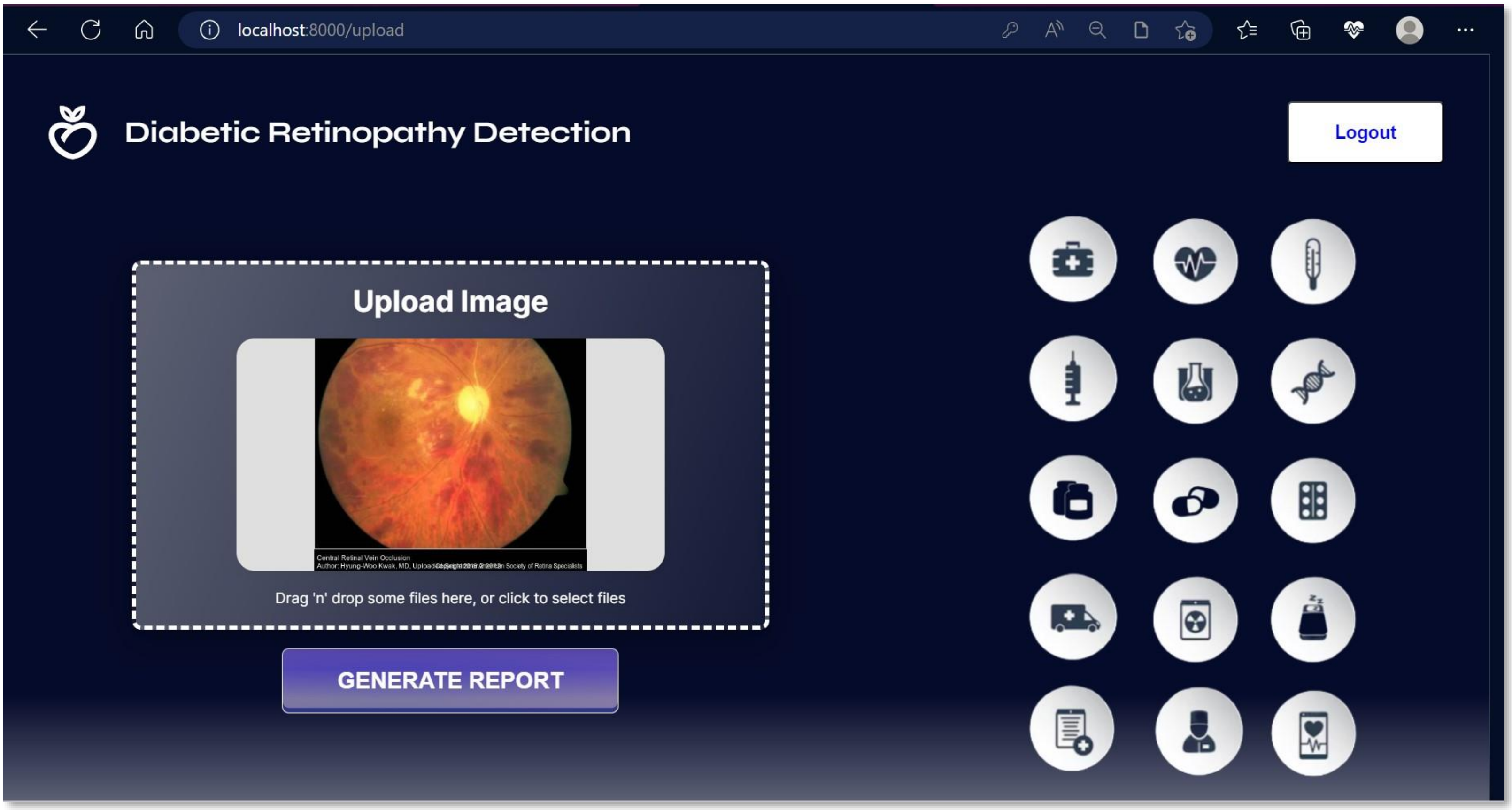
9.1 SIGNUP PAGE



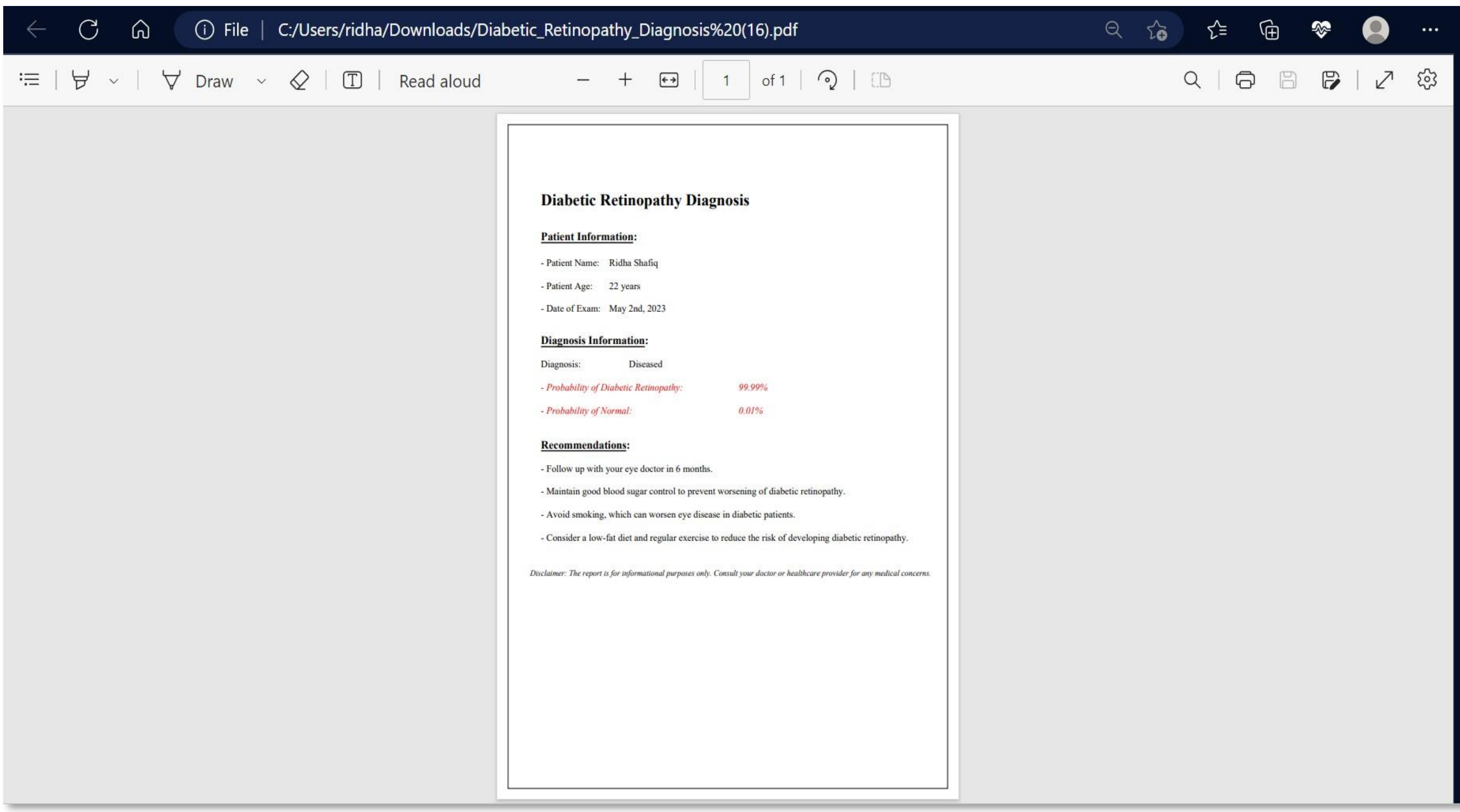
9.2 SIGNIN PAGE



9.3 UPLOAD PAGE



9.4 REPORT GENERATED



Diabetic Retinopathy Diagnosis

Patient Information:

- Patient Name: Ridha Shafiq

- Patient Age: 22 years

- Date of Exam: May 2nd, 2023

Diagnosis Information:

Diagnosis: Diseased

- Probability of Diabetic Retinopathy: 99.99%

- Probability of Normal: 0.01%

Recommendations:

- Follow up with your eye doctor in 6 months.

- Maintain good blood sugar control to prevent worsening of diabetic retinopathy.

- Avoid smoking, which can worsen eye disease in diabetic patients.

- Consider a low-fat diet and regular exercise to reduce the risk of developing diabetic retinopathy.

Disclaimer: The report is for informational purposes only. Consult your doctor or healthcare provider for any medical concerns.

Diabetic Retinopathy Diagnosis

Patient Information:

- Patient Name: Ridha Shafiq

- Patient Age: 22 years

- Date of Exam: May 2nd, 2023

Diagnosis Information:

Diagnosis: Diseased

- Probability of Diabetic Retinopathy: 99.99%

- Probability of Normal: 0.01%

Recommendations:

- Follow up with your eye doctor in 6 months.

- Maintain good blood sugar control to prevent worsening of diabetic retinopathy.

- Avoid smoking, which can worsen eye disease in diabetic patients.

- Consider a low-fat diet and regular exercise to reduce the risk of developing diabetic retinopathy.

Disclaimer: The report is for informational purposes only. Consult your doctor or healthcare provider for any medical concerns.

Chapter 10

Conclusion and Future Work

CHAPTER 10

CONCLUSION AND FUTURE WORK

In conclusion, the diabetic retinopathy web application developed in this thesis has the potential to significantly improve the diagnosis and treatment of diabetic retinopathy, particularly in rural areas of Pakistan. By using a private dataset from a renowned eye trust hospital, the application has been trained to accurately diagnose different stages of diabetic retinopathy, which can lead to early detection and treatment, and ultimately prevent blindness.

The Xception model used in the application has proved to be effective in image classification and can be further improved by incorporating more advanced machine learning techniques such as deep reinforcement learning and adversarial training. Additionally, the application can be integrated with telemedicine and mobile health technologies to reach a wider population and provide remote healthcare services to those living in remote or underserved areas.

Future work can focus on expanding the application to other countries and ethnic groups to improve the accessibility and inclusivity of healthcare services. Furthermore, the application can be integrated with electronic medical records to provide a comprehensive and longitudinal view of patient care, enabling better monitoring and treatment of diabetic retinopathy. Finally, the application can be further developed to incorporate additional features such as a patient portal that allows individuals to view their retinal images and diagnostic reports. This can help to increase patient engagement and encourage them to take an active role in managing their health.

Overall, the diabetic retinopathy web application has the potential to significantly improve the diagnosis and treatment of diabetic retinopathy, and further research and development can lead to significant improvements in healthcare outcomes for individuals affected by this condition.

Chapter 11

References and Work Cited

Chapter 11

REFERENCES AND WORK CITED

[1]. Qureshi, I.; Ma, J.; Abbas, Q. Diabetic retinopathy detection and stage classification in eye fundus images using active deep learning. Multimed. Tools Appl. 2021, 80, 11691–11721.

[2]. Abbas Q, Fondon I, Sarmiento A, Jiménez S, Alemany P (2017) Automatic recognition of severity level for diagnosis of diabetic retinopathy using deep visual features. Med Biol Eng Comput 55(11):1959–1974.

[3]. Ishtiaq U, Kareem SA, Abdullah ERMF, Mujtaba G, Jahangir R, Ghafoor HY (2019) Diabetic retinopathy detection through artificial intelligent techniques: a review and open issues. Multimed Tools Appl 79:1–44.

[4]. Guo S, Li T, Kang H, Li N, Zhang Y, Wang K (2019) L-Seg: An end-to-end unified framework for multilesion segmentation of fundus images. Neurocomput 349:52–63.

[5]. Mansour RF (2018) Deep-learning-based automatic computer-aided diagnosis system for diabetic retinopathy. Biomed Eng Lett 8(1):41–57.

[6]. Kermany DS, Goldbaum M, Cai W, Valentim CCS, Liang H, Baxter SL, McKeown A et al (2018) Identifying medical diagnoses and treatable diseases by image-based deep learning. Cell 172(5):1122–1131.

[7]. Li Z, Keel S, Liu C, He Y, Meng W, Scheetz J, Lee PY et al (2018) An automated grading system for detection of vision-threatening referable diabetic retinopathy on the basis of color fundus photographs. Diab care 41(12):2509–2516.

[8]. Lin J, Yu L, Weng Q, Zheng X (2019) Retinal image quality assessment for diabetic retinopathy screening: a survey. Multimed Tools Appl 79:1–27.

[9]. Orlando JI, Prokofyeva E, del Fresno M, Blaschko MB (2018) An ensemble deep learning-based approach for red lesion detection in fundus images. Comput Methods Prog Biomed 153:115–127.

[10]. Pang H, Luo C, Wang C (2018) Improvement of the application of diabetic

retinopathy detection model. Wirel Pers Commun 103(1):611–624.

[11]. Quellec G, Charrière K, Boudi Y, Cochener B, Lamard M (2017) Deep image mining for diabetic retinopathy screening. Med Image Anal 39:178–193.

TURNITIN REPORT

ic_Retinopthy_Detection_Web-Based_Application_-_Retina_Doc-2.pdf

ORIGINALITY REPORT

15%	10%	7%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	link.springer.com	Internet Source	2%
2	Submitted to Limerick Institute of Technology	Student Paper	1%
3	eprints.utar.edu.my	Internet Source	1%
4	ir-library.ku.ac.ke	Internet Source	1%
5	Submitted to Higher Education Commission Pakistan	Student Paper	1%
6	Submitted to University of East London	Student Paper	1%
7	www.coursehero.com	Internet Source	1%
8	elibrary.mec.edu.om	Internet Source	<1%

Submitted to University of Huddersfield

9	Student Paper	<1 %
10	assets.publishing.service.gov.uk Internet Source	<1 %
11	www.mdpi.com Internet Source	<1 %
12	prr.hec.gov.pk Internet Source	<1 %
13	Submitted to University of the West Indies Student Paper	<1 %
14	air.ashesi.edu.gh Internet Source	<1 %
15	Submitted to Liverpool John Moores University Student Paper	<1 %
16	Submitted to University of Central England in Birmingham Student Paper	<1 %
17	Figueiredo, I.N., S. Kumar, C.M. Oliveira, J.D. Ramos, and B. Engquist. "Automated lesion detectors in retinal fundus images", Computers in Biology and Medicine, 2015. Publication	<1 %
18	Submitted to Sheffield Hallam University Student Paper	<1 %

19	Submitted to Universiti Selangor Student Paper	<1 %
20	Submitted to University of Greenwich Student Paper	<1 %
21	digitalcommons.du.edu Internet Source	<1 %
22	Submitted to University of Cape Town Student Paper	<1 %
23	cs229.stanford.edu Internet Source	<1 %
24	Submitted to Coventry University Student Paper	<1 %
25	Natasha Shaukat, Javeria Amin, Muhammad Imran Sharif, Muhammad Irfan Sharif, Seifedine Kadry, Lukas Sevcik. "Classification and Segmentation of Diabetic Retinopathy: A Systemic Review", Applied Sciences, 2023 Publication	<1 %
26	Submitted to University of Wollongong Student Paper	<1 %
27	gsnb.rutgers.edu Internet Source	<1 %
28	www.researchgate.net Internet Source	<1 %

29	Muhammad Mateen, Junhao Wen, Mehdi Hassan, Nasrullah Nasrullah, Song Sun, Shaukat Hayat. "Automatic Detection of Diabetic Retinopathy: A Review on Datasets, Methods and Evaluation Metrics", IEEE Access, 2020 Publication	<1 %
30	Nurul Najihah A'bas, Sarni Suhaila Rahim, Mohamad Lutfi Dolhalit, Wan Sazli Nasarudin Saifudin et al. "Development and Usability Testing of a Consultation System for Diabetic Retinopathy Screening", International Journal of Advanced Computer Science and Applications, 2021 Publication	<1 %
31	journals.plos.org Internet Source	<1 %
32	repositorio.utad.pt Internet Source	<1 %
33	www.slideshare.net Internet Source	<1 %
34	www.tandfonline.com Internet Source	<1 %
35	"Advances in Computing and Data Sciences", Springer Science and Business Media LLC, 2021 Publication	<1 %

36	Munish Khanna, Law Kumar Singh, Shankar Thawkar, Mayur Goyal. "Deep learning based computer-aided automatic prediction and grading system for diabetic retinopathy", Multimedia Tools and Applications, 2023 Publication	<1 %
37	unsworks.unsw.edu.au Internet Source	<1 %
38	"Artificial Intelligence and Machine Learning for Digital Pathology", Springer Science and Business Media LLC, 2020 Publication	<1 %
39	"Electronic Systems and Intelligent Computing", Springer Science and Business Media LLC, 2020 Publication	<1 %
40	Ayoub Skouta, Abdelali Elmoufidi, Said Jai-Andaloussi, Ouail Ouchetto. "Deep learning for diabetic retinopathy assessments: a literature review", Multimedia Tools and Applications, 2023 Publication	<1 %
41	Qaisar Abbas, Irene Fondon, Auxiliadora Sarmiento, Soledad Jiménez, Pedro Alemany. "Automatic recognition of severity level for diagnosis of diabetic retinopathy using deep visual features", Medical & Biological Engineering & Computing, 2017 Publication	<1 %

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		