DETECTION & CLASSIFICATION OF DIABETIC RETINOPATHY USING INCEPTION V3 AND XCEPTION ARCHITECTURES

A PROJECT REPORT

Submitted by

BHARGHAVI.J	**211419104041**
DEEPTHI.H	**211419104052**
MATHUMITA.B	**211419104164**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE

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

APRIL 2023

BONAFIDE CERTIFICATE

Certified that this project report **"DETECTION & CLASSIFICATION OF DIABETIC RETINOPATHY USING INCEPTION V3 AND XCEPTION ARCHITECHTURES"** is the bonafide work of "**BHARGHAVI.J (211419104041), DEEPTHI.H (211419104052) and MATHUMITA.B (211419104164)"** who carried out the project work under Ms. **V. SATHIYA** supervision.

SIGNATURE	**SIGNATURE**
Dr. L. JABASHEELA, M.E., Ph.D.,	**Ms. V. SATHIYA**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
	ASSOCIATE PROFESSOR
DEPARTMENT OF CSE,	DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE,	PANIMALAR ENGINEERING COLLEGE,
NASARATHPETTAI,	NASARATHPETTAI,
POONAMALLEE,	POONAMALLEE,
CHENNAI- 600 123.	CHENNAI- 600 123.

Certified that the above candidate(s) was/ were examined in the Anna University Project Viva-Voce Examination held on...........................

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION BY THE STUDENT

We BHARGHAVI.J (211419104041), DEEPTHI.H (211419104052) and MATHUMITA.B (211419104164) hereby declare that this project report titled “DETECTION AND CLASSIFICATION OF DIABETIC RETINOPATHY USING INCEPTION V3 AND XCEPTION ARCHITECHTURES” , under the guidance of Ms. V. SATHIYA (Associate Professor) is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

DECLARATION BY THE STUDENT

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

We express our sincere thanks to our Directors **Tmt.C.VIJAYARAJESWARI**, **Dr.C.SAKTHI KUMAR,M.E.,Ph.D** and **Dr.SARANYASREE SAKTHI KUMAR B.E.,M.B.A.,Ph.D.,** for providing us with the necessary facilities to undertake this project.

We also express our gratitude to our Principal **Dr. K. MANI, M.E., Ph.D.,** who facilitated us in completing the project.

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

We would like to thank my guide **Ms. V. SATHIYA** and all the faculty members of the Department of CSE for their advice and encouragement for the successful completion of the project.

BHARGHAVI. J (211419104041),

DEEPTHI. H (211419104052),

MATHUMITA. B (211419104164)

ABSTRACT

Diabetic Retinopathy (DR) is a common disease among those affected by diabetes that occurs when there is damage to the retina. This damage typically occurs when there is too much sugar in the blood, which can cause blockage of the tiny blood vessels that nourish the retina by cutting off its blood supply. As a result, the eye attempts to grow new blood vessels. But these new blood vessels are either poorly developed or weak. So, it can be leaked out easily. Even at its most severe, it may result in permanent blindness in patients who have had DR for a long time. Therefore, it is necessary to diagnose these patients very soon to mitigate the severe impact of DR. Several methods were proposed earlier to identify this disease using machine learning algorithms, image processing, and so on. Diabetic retinopathy detection contains three steps -pre-processing of colour fundus images, diagnostic feature extraction and classification of DR. In this work, a Convolutional Neural Network (CNN)-based pre-trained transfer learning algorithm is used to speed up the diagnosis of the types of DR using fundus photography of retinal images of patients. The images are of three different classes, which define the types of DR. We used Inception V3 for DR detection to detect whether a person has DR or not, and Xception for DR classification to classify what type of DR it is. As a result, our research can help DR patients reduce their chances of becoming blind for life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVATIONS

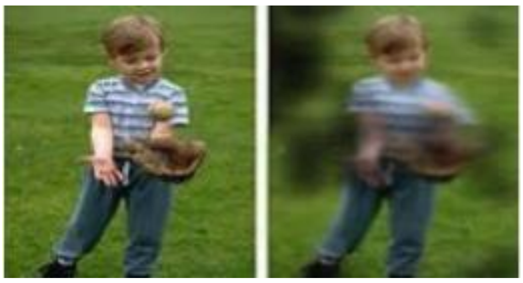
SHORTCUT	ABBREVATION	PAGE NO.
DR	Diabetic Retinopathy	7
MF	Median Filter	21
AHE	Adaptive Histogram Equalization	22

CHAPTER 1

INTRODUCTION

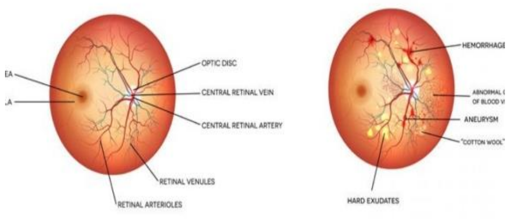
1.1. DIABETIC RETINOPATHY

Diabetic retinopathy is a condition that may occur in people who have diabetes. It causes progressive damage to the retina, the light-sensitive lining at the back of the eye. Diabetic retinopathy is a serious, sight-threatening complication of diabetes. Diabetes interferes with the body’s ability to use and store sugar (glucose). The disease is characterized by too much sugar in the blood, which can cause damage throughout the body, including the eyes. Over time, diabetes damages small blood vessels throughout the body, including the retina. Diabetic retinopathy occurs when these tiny blood vessels leak blood and other fluids. This causes the retinal tissue to swell, resulting in cloudy or blurred vision.



Normal Vision *Vision with Diabetic Retinopathy*

Fig1.1 *Vision with diabetic retinopathy*



Normal Retina *Diabetic Retinopathy*

Fig 1.2 *Normal retina and Diabetic retinopathy*

CAUSES OF DIABETIC RETINOPATHY

Diabetic retinopathy occurs when the sugar content blocks the small blood vessels located in the retina. As the area of the retina responsible for clear vision, these damaged blood vessels can cause vision loss. A portion of the retina that allows us to see colour and fine detail leaks a fluid.

The fluid causes the macula to swell, resulting in blurred vision. In an attempt to improve blood circulation in the retina, new blood vessels may form on its surface. These fragile, abnormal blood vessels can leak blood into the back of the eye and block vision.

WHAT HAPPEN IF WE DON’T TAKE PROPER MEASURES

When people with diabetic retinopathy don’t take proper measures to control the sugar in their blood, a fluid called macular edema accumulates in the lens that controls focusing. This changes the curvature of the lens, leading to changes in vision. However, once blood sugar levels are controlled, the lens usually returns to its original shape and vision improves. Patients with diabetes who can control their blood sugar levels will slow the onset and progression of diabetic retinopathy. One can help prevent or slow the development of diabetic retinopathy by:

- Taking your prescribed medication
- Sticking to your diet
- Exercising regularly
- Controlling high blood pressure
- Avoiding alcohol and smoking

1.1.1. TYPES OF DIABETIC RETINOPATHY:

Diabetic retinopathy (DR) is classified into two ways. They are Non-Proliferative diabetic retinopathy (NPDR) and Proliferative diabetic retinopathy (PDR).

Non-proliferative diabetic retinopathy (NPDR) is the early stage of disease in which symptoms will be mild or non-existent. In NPDR, the blood vessels in the retina are weakened. Tiny bulges in the blood vessels, called microaneurysms, may leaks fluid into the retina. This leakage may lead to swelling of the macula.

Numerous symptoms, such as hemorrhages (H), microaneurysms (MA), cotton wool (CWS), soft exudates (SE), or hard exudates (HE), may be present in NPDR as shown in figure 1.3.

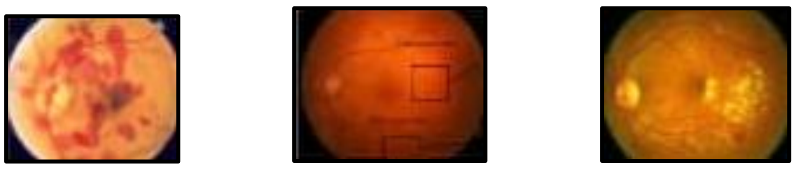


Figure 1.3 *Types of non-proliferative diabetic retinopathy*

The more severe type of the disease is proliferative diabetic retinopathy. At this stage, circulation problems deprive the oxygen in the retina. As a result, the retina and the vitreous, the gel-like fluid that fills the back of the eye, can start to develop new, delicate blood vessels. Vision may become blurry as a result of the new blood cells leaking into the vitreous.

The development of glaucoma and the detachment of the retina as a result of the formation of scar tissue are additional proliferative diabetic retinopathy side effects. An eye condition known as glaucoma causes the optic nerve to progressively deteriorate. Proliferative diabetic retinopathy is characterised by the development of new blood vessels in the region of the eye that drain fluid from the eye. This causes the eye pressure to significantly increase, which harms the optic nerve. Proliferative diabetic retinopathy can lead to serious vision loss and even blindness if left untreated.

1.1.2. STAGES OF DIABETIC RETINOPATHY

There are different stages of DR that affect a person with diabetes and are classified as mild, moderate, and severe as shown in figure. 1.4.

In the case of **mild diabetic retinopathy**, a small portion of the retinal blood vessels swells like a balloon, called a microaneurysm. It might consequently leak fluid into the retina, which could lead to retinal damage.

In the case of moderate **diabetic retinopathy**, as the condition worsens, blood vessels may swell even more and lose their ability to carry blood.

In case of severe **diabetic retinopathy**, the retina's blood flow is disrupted, and the blood vessels are harmed.

Furthermore, the formation of new blood vessels within the retina at this stage of DR is the most crucial one. These vessels cause bleeding and leakage, harm the tissues and could cause them to contract and induce retinal detachment.

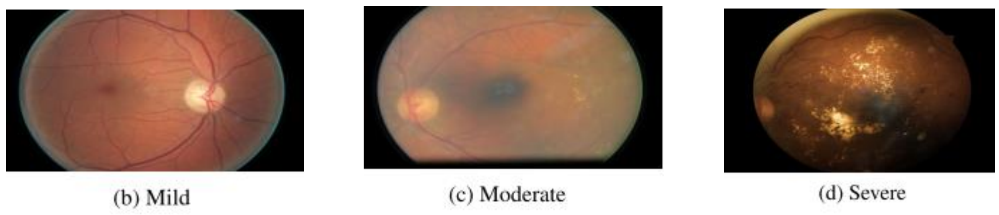


Figure 1.4 *Stages of diabetic retinopathy*

People who come under the following categories are most likely have the chances of developing diabetic retinopathy:

DIABETES: Diabetic retinopathy occurs in people with type 1 or type 2 diabetes. A person's risk of developing diabetic retinopathy increases with time spent living with the disease, especially if it is not well controlled.

PREGNANCY: Pregnant women face a higher risk for developing diabetes and diabetic retinopathy. If a woman develops gestational diabetes, she has a higher risk of developing diabetes retinopathy as she ages

RACE: Hispanics and African Americans are at greater risk of developing diabetic retinopathy.

OTHER MEDICAL CONDITIONS: Diabetic retinopathy is more common among individuals with other medical conditions like high blood pressure or high cholesterol.

Ways to diagnose Diabetic Retinopathy

Diabetic retinopathy can be diagnosed through a comprehensive eye examination. Evaluating the retina and the macula may include the following:

- ➢ Patient history to determine vision difficulties, presence of diabetes, and other general health concern which may have the chance of affecting the vision.
- ➢ Visual acuity measurements to determine how much central vision has been affected because of the diabetic retinopathy
- ➢ Refraction to determine if a new eyeglass prescription is needed
- ➢ Evaluation of the ocular structures, including the evaluation of the retina through a dilated pupil
- ➢ Determining the pressure inside the eye

Supplemental tests:

- Retinal photography or tomography to document the current status of the retina
- Fluorescent angiography or evaluate abnormal blood vessel growth

1.2 PROBLEM DEFINITION

The main objective of the project is to accurately detect whether a person has diabetic retinopathy, so that we can prevent the patients from permanent loss of vision and stop blindness before it is too late. Diabetes retinopathy has no symptoms in its early stages. Gradually, some people notice changes in their vision in the later phases, such as difficulty reading or not being able to see faraway objects.

The American Optometric Association (AOA) conducted the 2018 American EyeQ Survey, which found that almost half of Americans were unaware of the existence of any outward signs of diabetic eye diseases. The same survey revealed that more than one in three Americans were unaware that the simplest way to tell whether a person's diabetes will result in blindness is through a thorough eye exam. For this reason, the AOA advises that all individuals who have diabetes undergo a detailed dilated eye examination at least every year. This incident demonstrates the fact that the majority of patients aren't even aware that they have Diabetic Retinopathy.

The primary goal of this research is to create a web-based application that recognises whether an individual has Diabetic Retinopathy (DR) or not. If a patient has Diabetic Retinopathy, this model will determine the severity and type of the Diabetic Retinopathy. Due to the fact that this app is online, anyone with internet access can easily use it to check if they have Diabetic Retinopathy by uploading retinal images as an input to the system. Two deep learning algorithms were used in this case to boost accuracy by 98%. Inception V3 is used for detection, and Xception is used for classification.

For detection, initially we classified the fundus images into two classes, where each class has a label named "diabetic retinopathy and non-diabetic retinopathy". We trained the Inception V3 model using these two classes to detect whether a person has Diabetic Retinopathy based on the input image (fundus image) given by that person**.**

For classification, this model classifies the images that have diabetic retinopathy into three more classes, and each of these classes has a label named "microaneurysm," "exudate”, or "haemorrhage". We trained the Xception model using these classes to determine what type of diabetic retinopathy a patient has based on the fundus image given by the patient.

CHAPTER 2 LITERATURE SURVEY

2.1. Harshit Kaushik,Dilbag Singh,Manjit Kaur, Hammam Alshazly, Atef Zaguia and Habib Hamam [1] developed three different sub-models of CNNs that are fed into a single meta-learner classifier for feature extraction. A data augmentation technique is also applied to improve the diversity of images in the dataset. Finally, the meta-learner classifier produced the diagnostic result as healthy (no DR) or unhealthy (DR).

2.2.Mohamed M. Abdel Salam, M. A. Zahran [2] proposed a new, accurate, and efficient algorithm based on the framework of multifractal analysis to classify and characterise any complex shapes and branching patterns found in physics and biology. This study provides strong evidence that multifractal analysis can be used as a screening tool for the early detection of retinal diseases. The work used a supervised machine learning method called the Support Vector Machine (SVM) algorithm to automate the diagnosis process and improve the resultant.

2.3.Mohamed M. Farag,Mariam Foud, and Amr T. Abdul-Hamid [3] presented a new CNN model based on DenseNet169 architecture integrated with CBAM as an additional component to be added for representational power enhancement. The proposed method demonstrated robust performance and comparable quality metrics while reducing the burden of space and time complexity. Furthermore, a 2-D Gaussian filter enhanced the fundus images' quality. Finally, they used INS to form a weighted loss function to tackle the class imbalance and improve the model's prediction across all.

2.4. Zhentao Gao, Jie Li, Jixiang Guo, Yuanyuan Chen, Zhang Yi, and JieZhong [4] investigated the automatic grading of DR using deep neural networks and proposed a novel dataset that is moderate in size and is annotated with a new labelling scheme. A pre-processing pipeline to change fundus images into a uniform format is also proposed.Visualization and analysis of the trained models provide insights into how the models make diagnoses using given fundus images and justify the diagnostic ability of the models from a different viewpoint. For clinical applications, the trained models are deployed on a cloud computing platform and provide pilot diagnostic services to several hospitals via the internet.

2.5. Syed Farooq Ali, Hamza Mustafa, Muhammad Bilal, and Muhammad Shehzad Hanif [5] proposed a novel ensemble technique for automated grading and classification of diabetic retinopathy that has been introduced and is built upon deep learning models, namely ResNet-50 and DenseNet-121. This paper presents a multi-stream deep neural network for classification and grading of diabetic retinopathy using EyePACS, Messidor-2, APTOS, and DDR datasets using 2, 3, and 5 categories. This model used two streams of inputs in the form of features extracted from the two deep networks.

2.6. Sharmin Majumder and Nasser Kehtarnavaz [6] proposed a multitasking deep neural network to classify all five stages of diabetic retinopathy from eye fundus images based on the DenseNET architecture. The two largest publicly available datasets of eye fundus images (the EyePACS and APTOS datasets) were used to train and evaluate the developed model. The outcomes demonstrate that the developed multitasking model produced the best performance.

2.7. Zubair Khan, Fiaz Gul Khan, Ahmad Khan,Zia Ur Rehman, Sajid Shah, Sehrish Qummar, Farman Ali, and Sangheon Pack [7] In this paper, architectural changes are made to the existing CNN to enhance the efficiency and accuracy for the classification of the Diabetic Retinopathy stages in colour fundus images and reduce the number of learnable parameters. Imbalanced versions of the Kaggle dataset are used to validate the performance measures of the proposed model.

2.8. Fahman Saeed, Muhammad Hussain, and Hatim A. Aboalsamh [8] created a deep learning-based automated system for grading retinal fundus images. The system was built on a pre-trained model using a two-stage transfer learning method. First, the CNN models are pre-trained on ImageNet. Second, the FC layers encode the high-level features relevant to the natural images and have a very large number of learnable parameters.

2.9. Xianglong Zeng, Haiquan Chen, Yuan Luo, and Wenbin Ye [9] developed a novel CNN model to automatically detect RDR based on the deep learning method. The proposed model has a Siamese-like architecture that accepts binocular fundus images as inputs and predicts the possibility of RDR for each eye by utilising the physiological and pathological correlations of both eyes.

2.10. Tieyuan Liu, Yi Chen, Hongjie Shen, Rupeng Zhou, Meng Zhang, Tonglai Liu and Jin Liu [10] research is based on a symmetric convolutional structure to detect different kinds of lesions in diabetic retinal images. The symmetrical convolutional structure can extract more complex lesion features and significantly improve detection performance.

CHAPTER 3
SYSTEM ANALYSIS

3.1. EXISTING SYSTEM

The algorithm put forward in the foundational research, "Hybrid Retinal Image Enhancement Algorithm for Diabetic Retinopathy Diagnosis using Deep Learning Model," can only determine whether or not a person has diabetic retinopathy. They have created hybrid modelling systems. One model is used to improve images, and another is used to identify diabetic retinopathy. The fundus images' quality has been improved using an improvement strategy. The primary goals of this work are to enhance retinal picture quality and increase accuracy using circle cropping. Here, DR detection is carried out following the image processing stage to enhance the features that were collected from the image. ResNet50, a deep learning model that offers a wealth of features to help with categorization, is the foundation for the feature extraction process. To increase image contrast and enhance image quality, they developed a hybrid image enhancement method. There are two functionalities in the model. They are,

- Cropping image from grayscale
- Circle crop and Gaussian bur

Furthermore, a deep learning network was created, whose function is to accurately classify DR and extract pertinent information.

Figure 5 shows the fundus image enhancement model. Retinal fundus images are sent to it, as seen in the figure. Foreground identification is the next operation it makes. The Gaussian blurring subblock receives the results afterward. It is the latter that is in charge of illuminating the retinal arteries. The network of modified Resnet50, which is in charge of deep learning feature extraction and classification, is then given the results of the processed image. The projected decision is sent to the end user as Resnet50's final output.

The algorithm takes two inputs: a threshold and an image. To separate the foreground from background, a level of intensity termed threshold is used. The program takes into account both RGB and grayscale images.

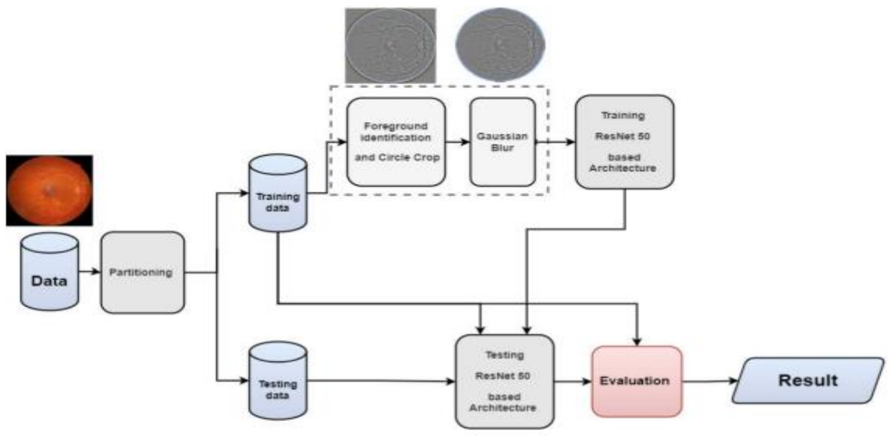

Foreground identification and Circle Crop
Gaussian Blur
Training ResNet 50 based Architecture
Training data
Data
Partitioning
Testing data
Testing ResNet 50 based Architecture
Evaluation
Result

Figure 3.1 *Conceptual fundus image enhancement*

The segmented image is returned in the same format as the input image. The mask separates the foreground in both grey and RGB images. The segmented photos are then output after that.

The quality and contrast are increased using a circle crop and the Gaussian blur method. The processed image is the algorithm's output.

The Resnet architecture serves as the foundation for the feature extraction and classification model proposed in this work, which is an ImageNet pre-trained neural network. The primary structure is depicted in Figure 3.2.

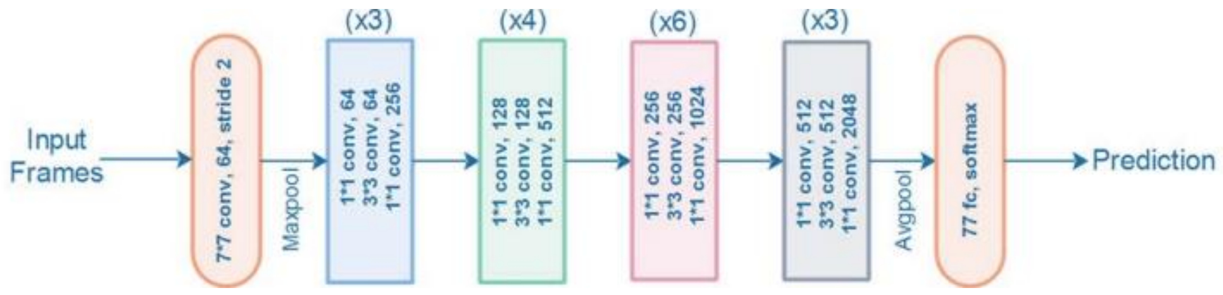


Figure 3.2 *ResNet-50 neural network*

The data set was obtained via Kaggle. For every subject in this dataset, there are pictures of both eyes. The pictures were taken using a variety of cameras under various lighting, weather, and resolution settings.

For training and testing purposes, the dataset is split into two sets, each of which contains images of the eyes taken from a separate patient. This dataset includes many fundus photographs for five classes of DR that were taken under various conditions.

The free DR screening portal EyePACS is the source of these fundus photos. The model is further tested using the MESSIDOR dataset of 1200 fundus pictures. Five categories—Normal, Mild, Moderate, Severe, and PDR—are employed in the search.
They used DRRNet, which is based on ResNet50 and designed to extract the features and classify them into five DR classes, to test our enhancement model.
The image enhancement stage starts the procedure, then the image proceeded to the neural network, they have executed the network with 40 epochs, patch size 8, learningrateis1e-4,and the optimization algorithm is Adam.
The experiments show that the measures of DR classification by using Kaggle original images without enhancement was about 73% accuracy, whereas after applying our enhancement model, the classification accuracy reached to 92%.

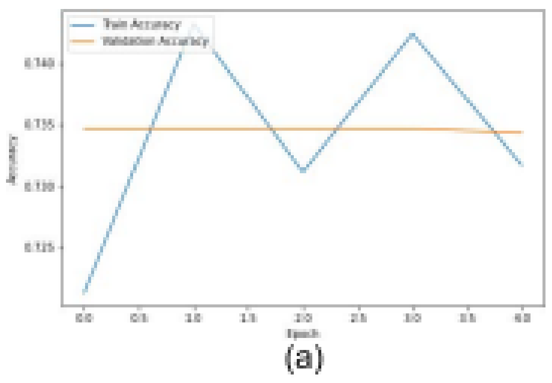


Figure 3.3 *Training accuracy without enhancement*

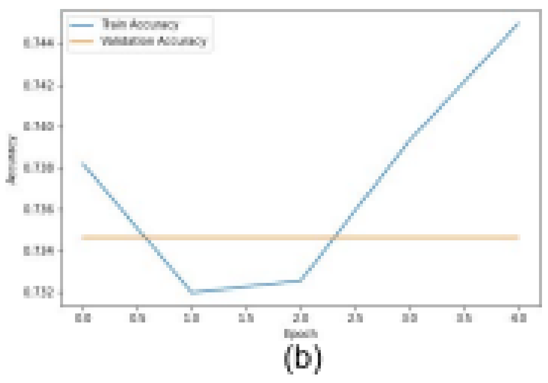


Figure 3.4 *Training accuracy with enhancement*

The training diagram is shown in Fig. 6. This shows that the suggested approach was effective in improving fundus image contrast while maintaining a high level of details and structure of lesions. It's also, successfully removed blurriness and noise from DR fundus images. It is evident from the experiments that the suggested technique enhanced fundus image quality more effectively, which improved the feature extraction and the predication accuracy than the state-of-the-art approaches.
A circular crop is applied on the image parallel with increasing contrast by gaussian blur algorithm. Finally, the resulted image is progressed as an input to a modified ResNet50 which achieved highest accuracy the state-of-the-art.

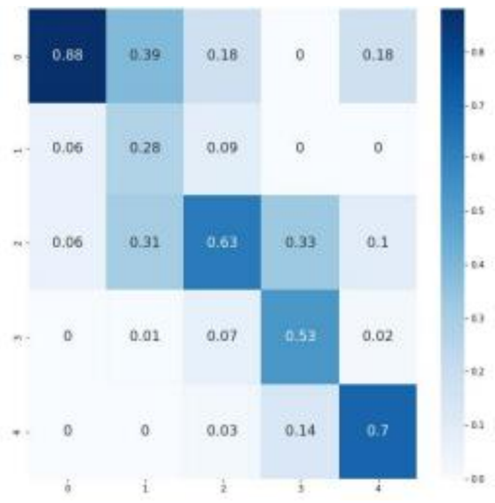


Figure 3.5 *Confusion matrix of existing system*

DISADVANTAGES OF EXISTING SYSTEM

- The existing system cannot determine the type and severity of diabetic retinopathy.
- The ResNet50 model was used for feature extraction and classification, which increased time consumption.
- The accuracy of the model is low (92%) when compared to the proposed model

3.2. PROPOSED SYSTEM

Our proposed methodology strongly emerged based on these key aspects of diseases In the proposed system we have used two models. They are,

- Inception V3
- Xception

We are using InceptionV3 for detection and Xception for classification.

3.2.1. DATASET:

IDRiD database is used to get the dataset. Each subject's two eyes are represented by photos in this dataset. Ophthalmologists, as was already mentioned, classify all images using the standard severity scale. Several cameras, with varying settings for lighting, situations, and resolutions, were used to take the pictures. For training and validation, the dataset is split into two sets, each of which contains images of the eyes taken from a separate patient. By altering copies of the existing dataset, the dataset will first be augmented to enhance the training dataset. Making small adjustments to the current dataset or creating new data using deep learning are both included.

The dataset will be loaded and pre-processed after the data has been gathered. The data is then divided into five categories, as shown in the table. 1 listed beneath

Class Name	Train	Validate	Total
Normal	34	16	50
Abnormal	34	16	50
Microaneurysm	28	8	36
Exudates	27	6	33
Haemorrhage	27	13	40

Table 3.1 *Distribution of classes in the dataset*

3.2.2. IMAGE PRE-PROCESSING

Improving the quality of the input photographs is a straightforward technique to boost our model's performance. By utilising the median filter (MF) to remove noise from the image and the adaptive histogram equalisation to increase the image's contrast, we were able to enhance the image's quality without sacrificing any data (AHE).

MEDIAN FILTER (MF): The median filter is used to remove noise during pre-processing. It is a filtering method used to take the noise out of the photos. Since the median filter is well known for maintaining edges during noise removal, it is extremely important in the field of image processing. The current system makes use of gaussiain blur. This may occasionally result in the loss of important image features. To improve the accuracy, we have therefore applied a median filter.

ADAPTIVE HISTOGRAM EQUALIZATION (AHE): once the picture has been cleaned up of the noise. The contrast of the image is enhanced using the adaptive histogram equalisation approach. It is a method of image pre-processing used to enhance the contrast of the picture. To redistribute the brightness values of the image, it computes a number of histograms, each one representing a different region of the image.

3.2.3. DETECTION AND CLASSIFICATION OF DIABETIC RETINOPATHY

After the photos have undergone adaptive histogram equalisation and the median filter in the preprocessing stage. The Inception V3 model will be trained to determine if a person has diabetic retinopathy or not using photos of normal and abnormal diabetic retinopathy images. Similarly, to train the Xception model for severity classification, fundus pictures with microaneurysm, exudates, and haemorrhage will be employed.
Under the current system, hybrid models are only utilised to determine whether or not a person has diabetic retinopathy (DR). However, we have employed hybrid models to more accurately detect and categorise diabetic retinopathy than the current system.

ADVANTAGES OF PROPOSED SYSTEM

- The proposed system can determine the specific type of DR in abnormal retinal images.
- The rate of classification accuracy is high due to image pre-processing techniques.
- The processing speed is high, so it consumes less computational time.

3.3. FEASIBILITY STUDY

3.3.1 OPERATIONAL FEASIBILITY

Operational feasibility is the measure of how well a proposed system solves the problems with the users.

Operational feasibility is dependent on human resources available for the project and involves projecting whether the system will be used if it is developed and implemented. The project is operationally feasible for the users as nowadays almost everyone are familiar with websites and digital technology.

3.3.2. ECONOMIC FEASIBILITY

Economic feasibility defines whether the expected benefit equals or exceeds the expected costs. It is also commonly referred to as cost/benefit analysis. The procedure is to determine the benefits and the savings expected from the system and compare them with the costs. A proposed system is expected to outweigh the costs.

This is a project with less cost for development. The system is easy to understand and use.

Therefore, there is no need to spend time on how to use the system. This system has the potential to grow by adding functionalities like using clinical dataset for training and testing rather than using dataset from open source. This might have the chance of increasing the accuracy even more in the coming future. Hence, the project could have economic benefits in the future.

3.3.3. TECHNICAL FEASIBILITY

Technical feasibility is carried out to determine whether the project is feasible in terms of software, hardware, personnel, and expertise, to handle the completion of the project. It considers determining resources for the proposed system.

As the system is developed using python, it is platform independent.

Therefore, the users of the system can have average processing capabilities, running on any platform. Since, this system is deployed on the internet using streamlit. Anyone can easily be able to access the system from anywhere through internet. The technology is one of the latest hence the system is also technically feasible.

3.4. HARDWARE REQUIREMENTS

System	Core i5 Processor
Hard Disk	500 GB
Monitor	15'' LED
Input Devices	Keyboard, Mouse
Ram	4 GB

Table3.2 *Hardware Requirements*

3.5. SOFTWARE REQUIREMENTS

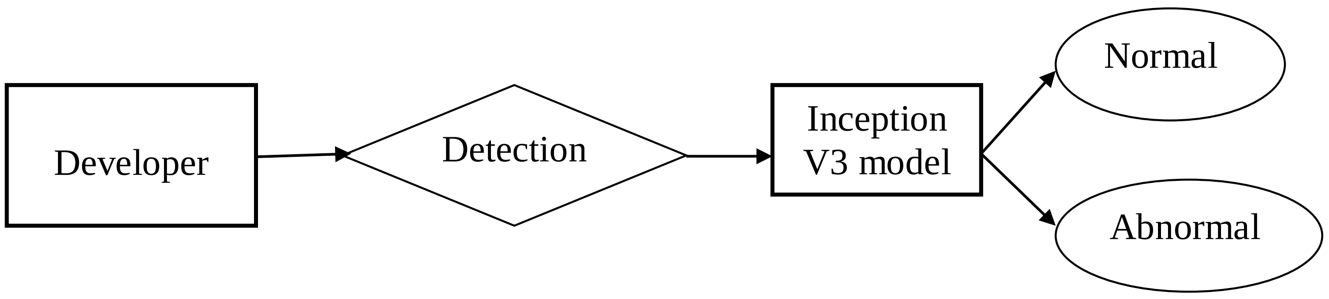
Operating System	Windows 10
Coding Language	Python 3.11
IDE	Visual Studio Code
Package Manager	PIP
Framework	Streamlit

Table3. 3 *Software Requirements*

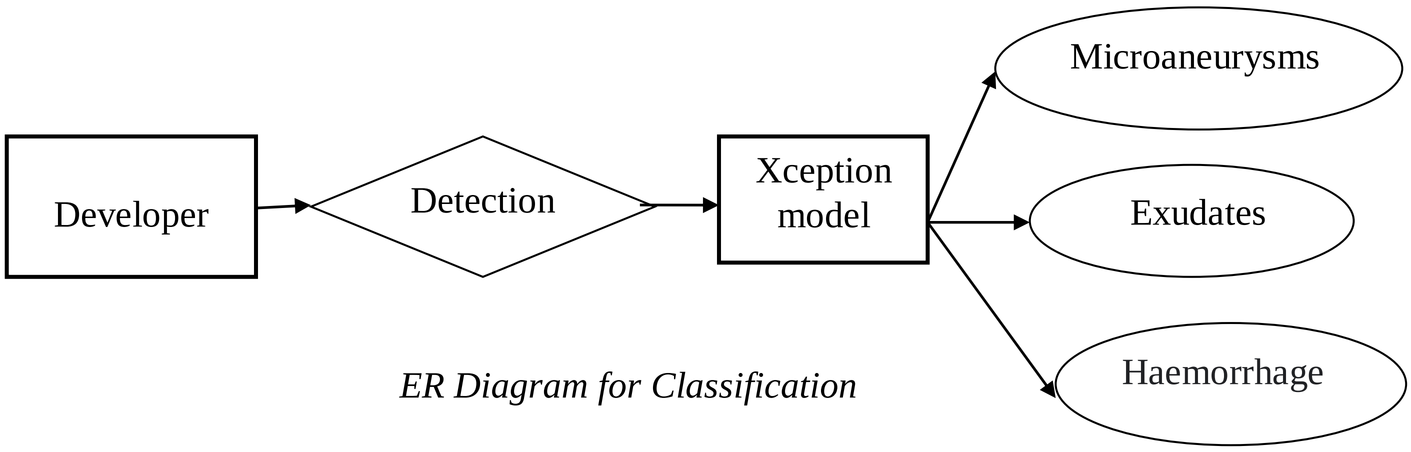
CHAPTER 4
SYSTEM DESIGN

4.1. ER DIAGRAM

Entity-Relationship model is referred to as ER model. It is a high-level data model. The data items and relationships for a given system are defined using this model. It creates the database's conceptual design. Also, it creates a very straightforward and Straightforward data view. In ER modelling, a diagram known as an entityrelationship diagram is used to represent the database structure.



ER Diagram for Detection

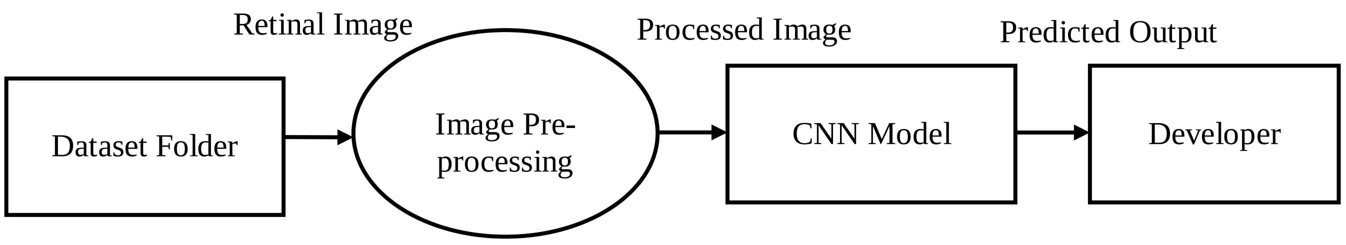


ER Diagram for Classification

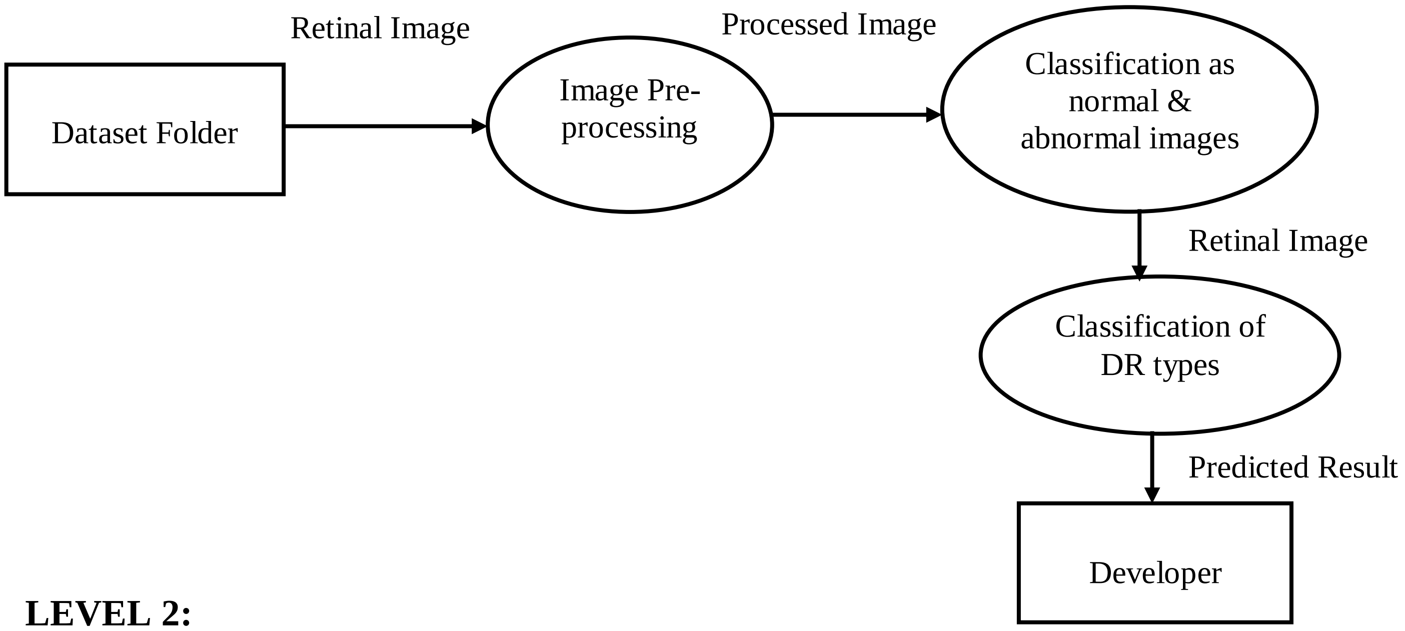
4.2 DATAFLOW DIAGRAM

A data flow diagram (DFD) is a picture that shows how information moves through a system or process. DFD said in process understanding so you can find possible issues, increase productivity, and create better processes. Data flow diagram symbols are standardized notations, such as rectangles, circles, and arrows. That shows the direction of data flow inside a system or process as well as its inputs, outputs, and storage places. From the high level data flow diagram to the low level data flow diagram, each level in this representation of the system's flow goes deeper.

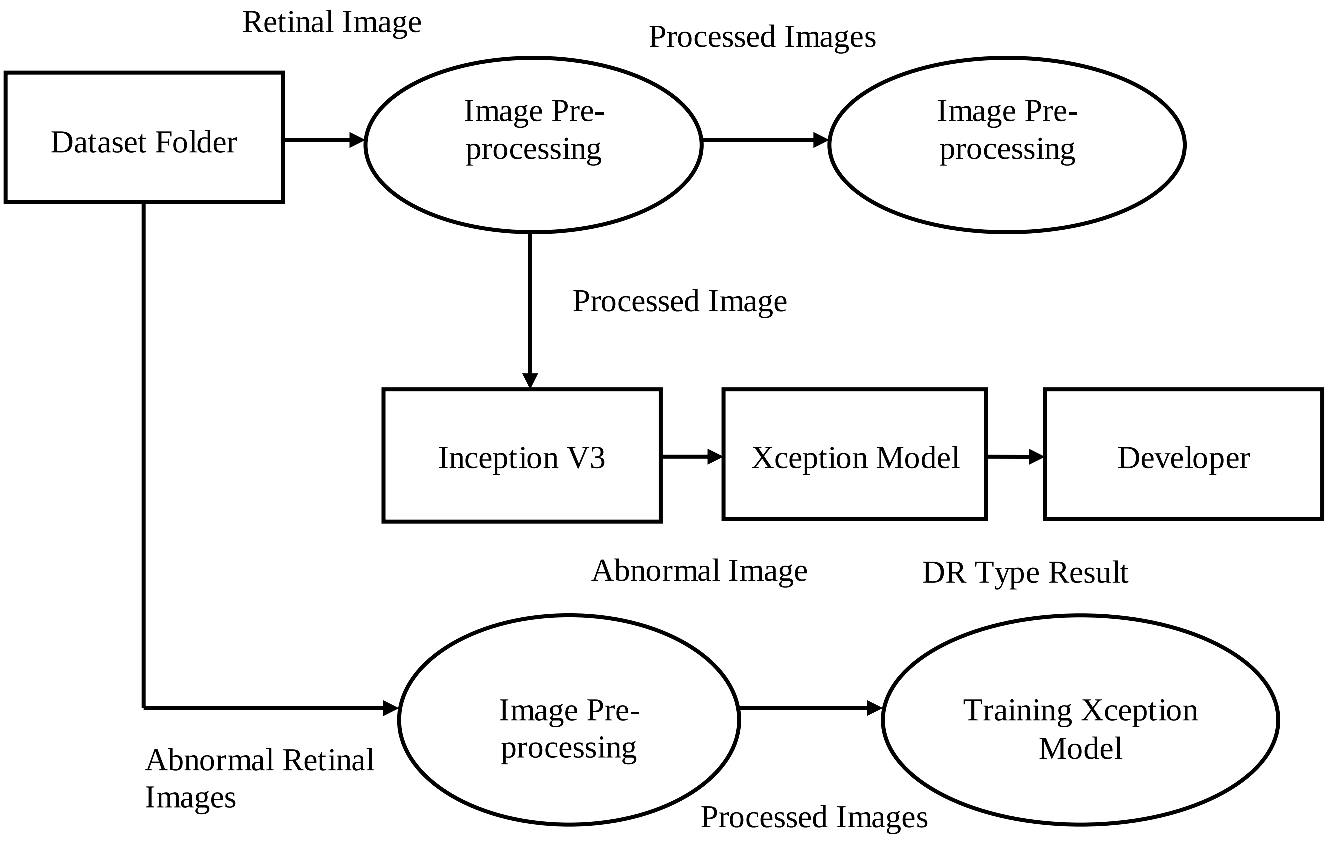
LEVEL 0:



LEVEL 1:



LEVEL 2:



4.3. UML DIAGRAMS

In order to better understand, update, maintain, or document information about the system, a UML diagram is a diagram based on the UML that aims to visually describe a system together with its primary players, roles, actions or classes. It is used to represent the linkages and structural elements of a complicated system.
There are two types of UML diagrams: structural diagrams and behavioural diagrams. Structure Diagrams includes Class Diagram, Component Diagram, Deployment Diagram, Object Diagram, Package Diagram, Profile Diagram, Composite Structure Diagram. Behavioral Diagrams includes Use Case Diagram, Activity Diagram, State Machine Diagram, Sequence Diagram, Communication Diagram, Interaction Overview Diagram.

4.3.1. USE CASE DIAGRAM

Use case diagrams provide a high-level overview of a system's capabilities. The interactions between the system and its actors are also depicted in these diagrams. Use-case diagrams show what the system does and how the actors utilise it, but they do not show how the system works within.

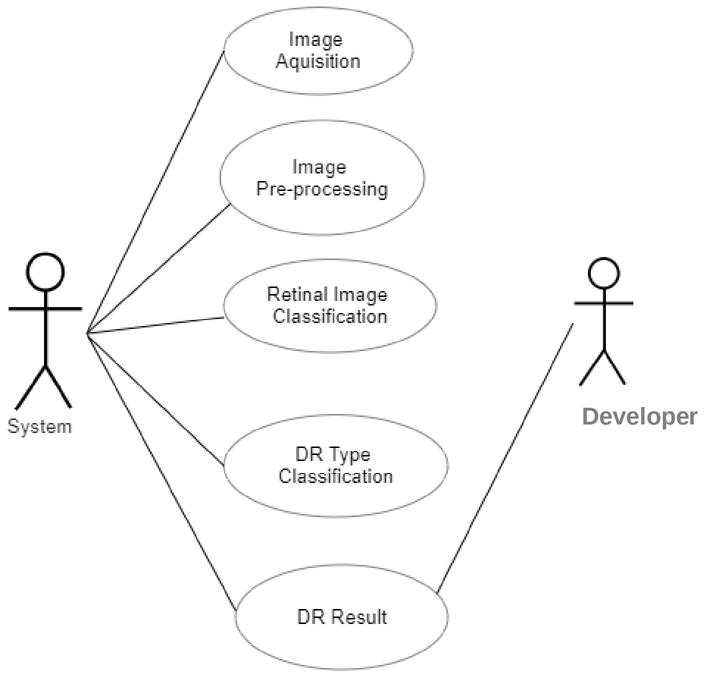


Figure 4.1 *Use case Diagram*

4.3.2. SEQUENCE DIAGRAM

Sequence diagrams depict interactions between classes as a time-based message exchange. Another name for them is event diagrams. A sequence diagram is an effective tool for visualizing and verifying different runtime scenarios. They can aid in system behaviour prediction and the identification of potential class responsibilities during the modelling of new systems.

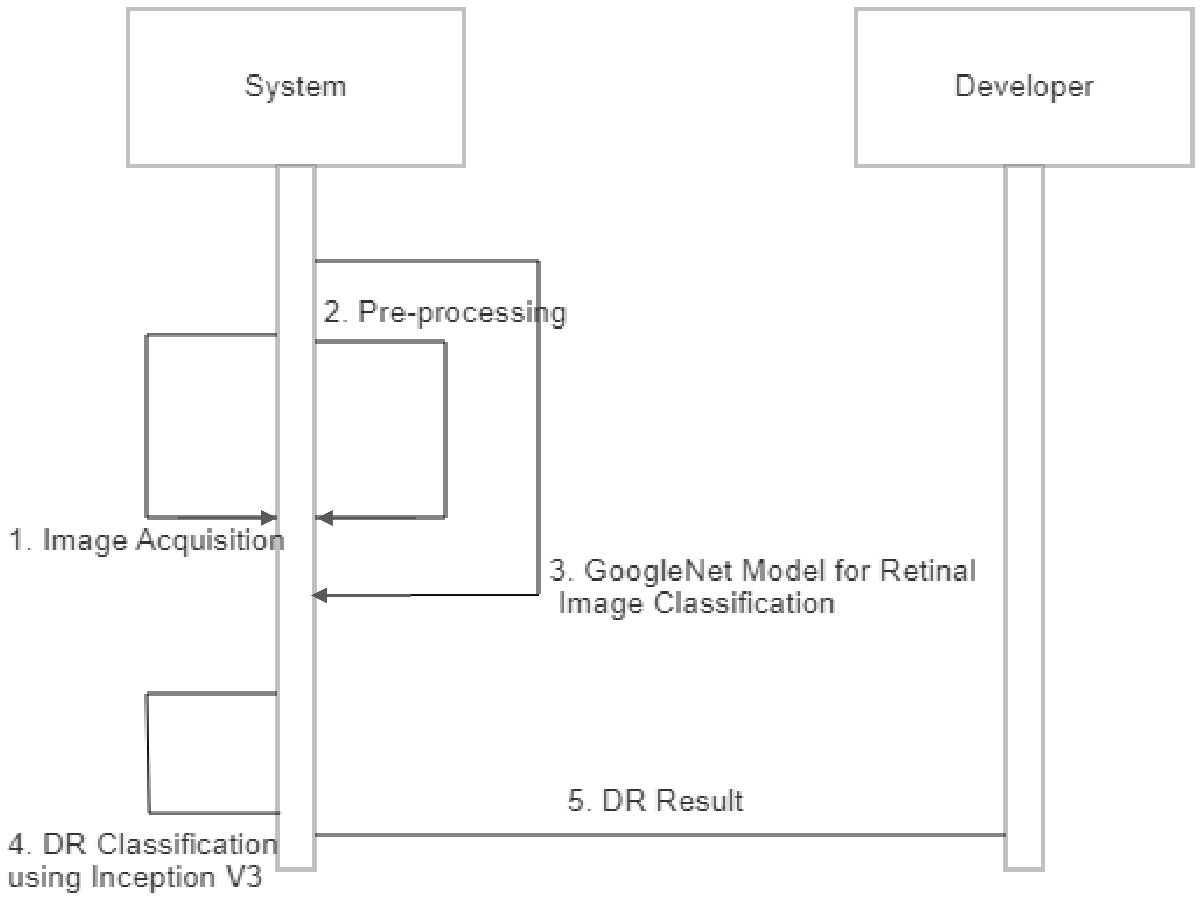

System
Developer
2. Pre-processing
1. Image Acquisition
3. GoogleNet Model for Retinal Image Classification
5. DR Result
4. DR Classification using Inception V3

Figure 4.2 *Sequence Diagram*

4.3.3. ACTIVITY DIAGRAM

An activity diagram is essentially a flowchart that shows how one activity leads to another. The action might be referred to as a system operation. One operation leads to the next in the control flow. This flow may be parallel, contemporaneous, or branched. Activity diagrams use many features, such as fork, join, etc., to cope with all types of flow control.

- ellipses represent actions
- diamonds represent decisions
- bars represent the start (split) or end (join) of concurrent activities
- a black circle represents the start (initial node) of the workflow
- an encircled black circle represents the end (final node)

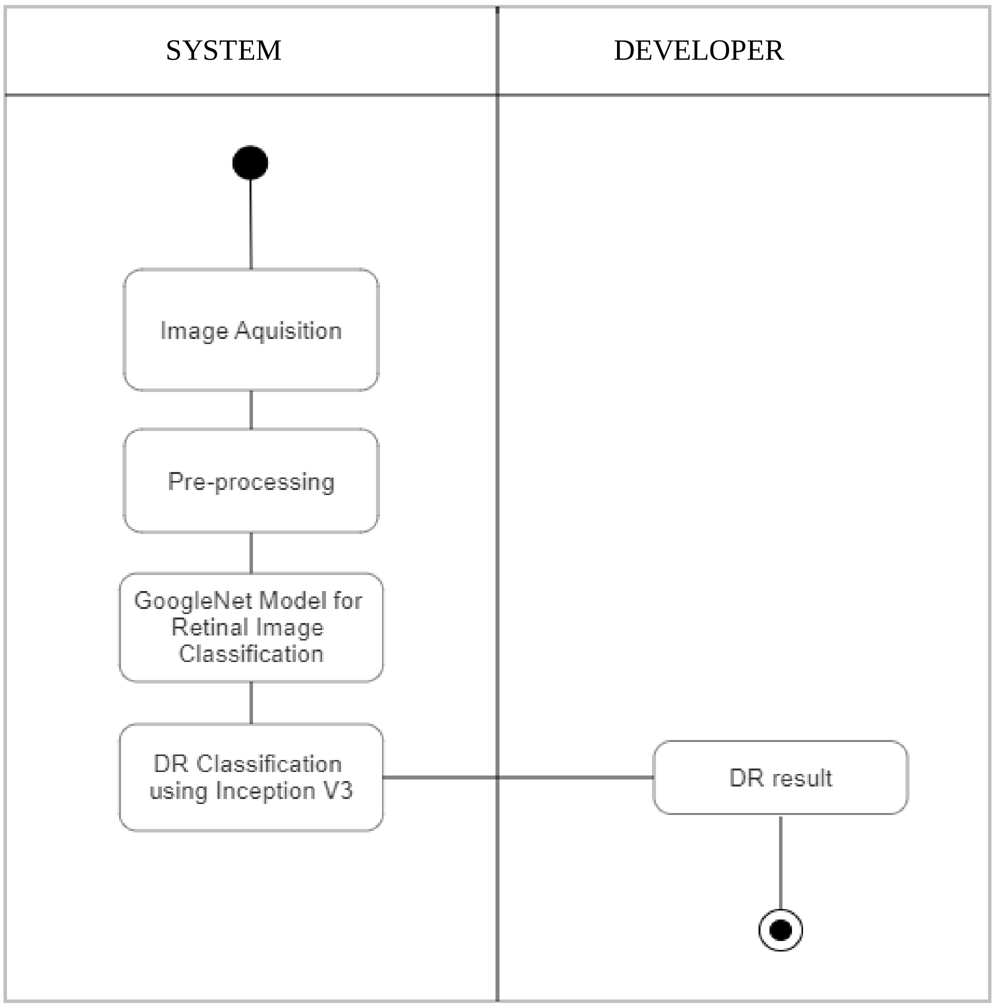

SYSTEM
DEVELOPER
Image Aquisition
Pre-processing
GoogleNet Model for Retinal Image Classification
DR Classification using Inception V3
DR result

Figure 4.3 *Activity Diagram*

CHAPTER 5

SYSTEM ARCHITECTURE

5.1 ARCHITECHTURE OVERVIEW

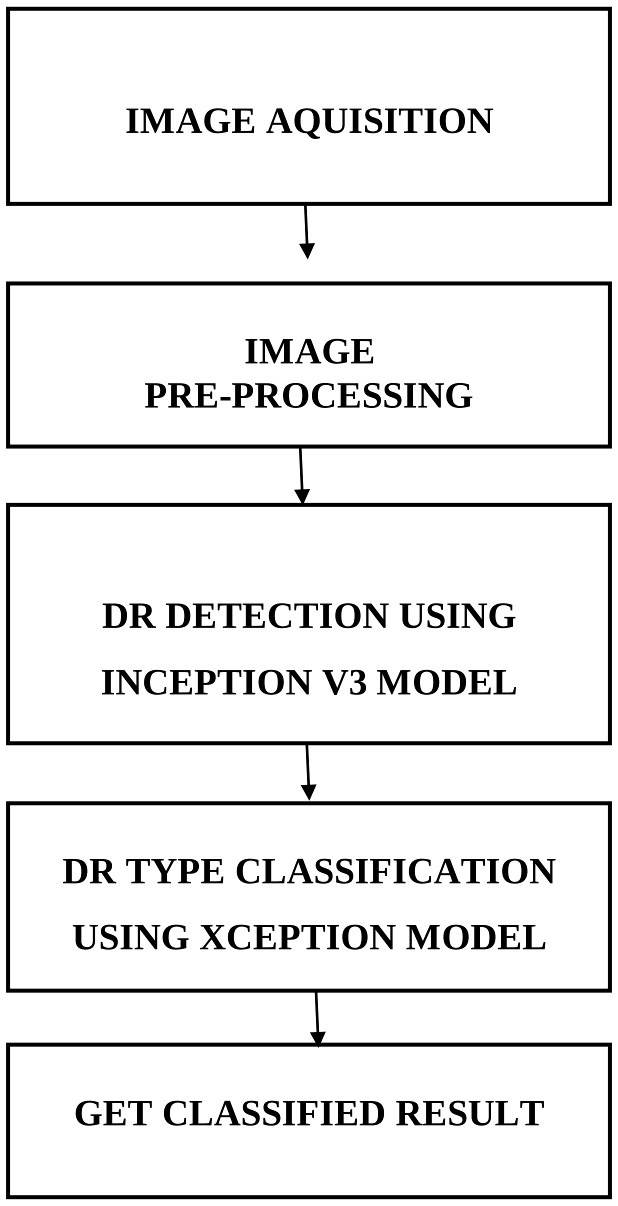


IMAGE AQUISITION
IMAGE PRE-PROCESSING
DR DETECTION USING INCEPTION V3 MODEL
DR TYPE CLASSIFICATION USING XCEPTION MODEL
GET CLASSIFIED RESULT

Figure 5.1 *Overview of the process*

The fundus images with DR and without DR will be collected and saved in different folders. Following image acquisition, the images will undergo preprocessing before being used to train the models. Once the pre-processing is completed. Normal and abnormal images are used to train the Inception V3 model. The next step is to create three folders and place the fundus images with hemorrhages, microaneurysms, and exudates inside each separate folder. Xception will be trained using these images, and its performance will be measured based on its accuracy.

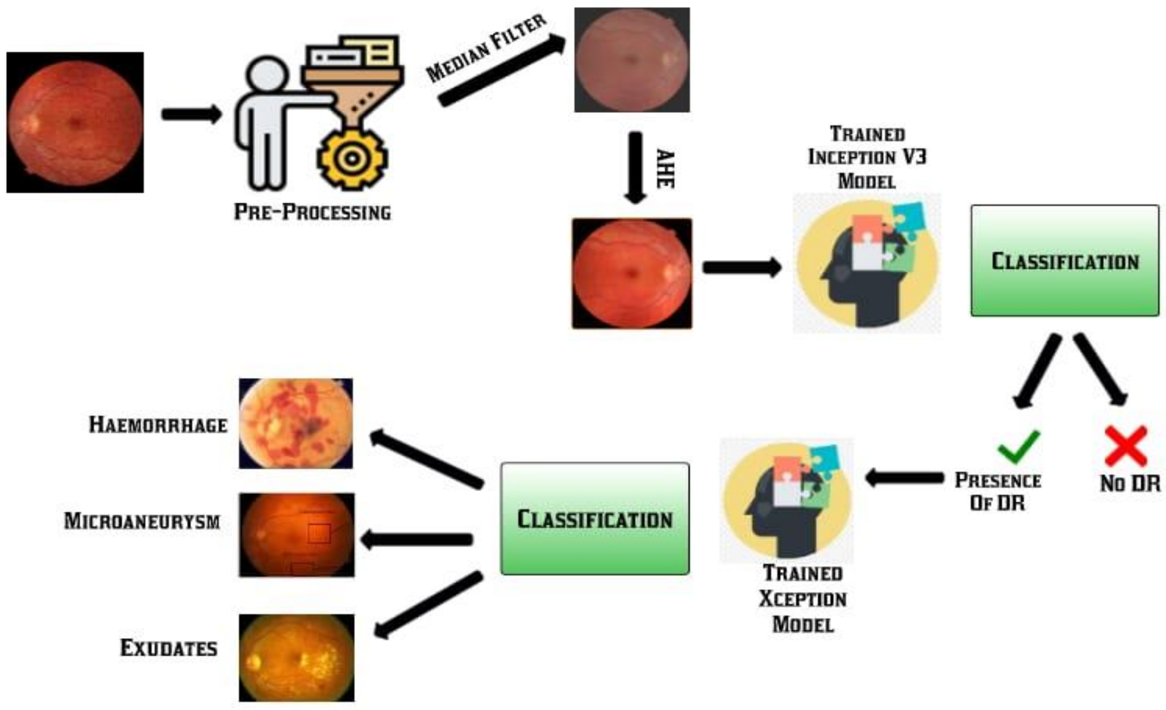


Figure 5.2 *System Architecture*

5.2. MODULE DESIGN SPECIFICATION

IMAGE ACQUISITION

In the context of image processing, image acquisition is typically defined as the method of extracting an image from a certain source, generally a hardware-based source. The IDRiD database is used to gather retinal images.

IMAGE-PREPROCESSING

The second stage is preprocessing, which includes image resizing and noise removal. The noise in the retinal image is eliminated using a median filter.

DIABETIC RETINOPATHY DETECTION

After preprocessing, features from retinal images are extracted and classified using the Inception V3 model. The input size for Inception V3 is 299 by 299 pixels with 42 layers. The proposed model was trained to create a classification prediction that divides the retinal image into normal and abnormal categories based on hyper-parameters like epochs, learning rate, dropout, and optimizer (ADAM) (diabetic retinopathy).

DIABETIC RETINOPATHY CLASSIFICATION

Using the Xception deep learning model, this module classified the images of diabetic retinopathy into three types using abnormal retinal images (DR). The proposed model was trained to build a classification prediction that categorizes the DR retinal image into three types, such as microaneurysms, hemorrhages, and exudates, based on hyper-parameters like epochs, learning rate, dropout, and optimizer (ADAM).

PERFORMANCE MEASURE

Finally, the effectiveness of the suggested model was measured in terms of accuracy, precision, recall, and F1 score performance metrics.

5.3. ALGORITHMS

5.3.1. ALGORITHM FOR DETECTION – INCEPTION V3

Inception V3 is an image classification deep learning model constructed using CNN. It is an enhanced version of the V1 model, the entry-level model from 2014. The model employs a number of network optimisation techniques for better alternative adaptation. Especially in contrast to the V1 Inception model and the V2 Inception model, the V3 model is more efficient and features a deeper network, but its speed is unaffected. Inception V3 uses auxiliary classifiers to regularise and is less computationally expensive.The 42-layer Inception V3 model, which was released in 2015, has a fairly low failure rate compared to its predecessors. Factorization into the Smaller Convolutions, Spatial Factorization into the Asymmetric Convolutions, Efficient Grid Size Reduction and Utility of Auxiliary Classifiers are the main improvements made to the Inception V3 model.

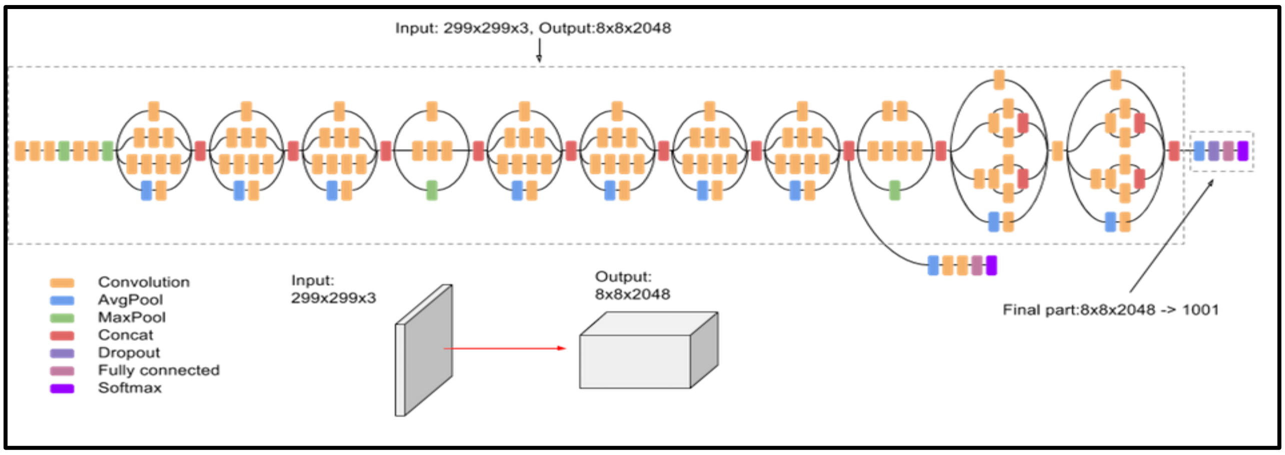


Figure 5.3 *Architecture of Inception V3* Model

ADVANTAGES OF INCEPTION V3 MODEL

- Higher efficiency
- Deeper network compared to the Inception V1 and V2 models
- Less expensive

5.3.1. 1. TRAINING AND VALIDATION ACCURACY

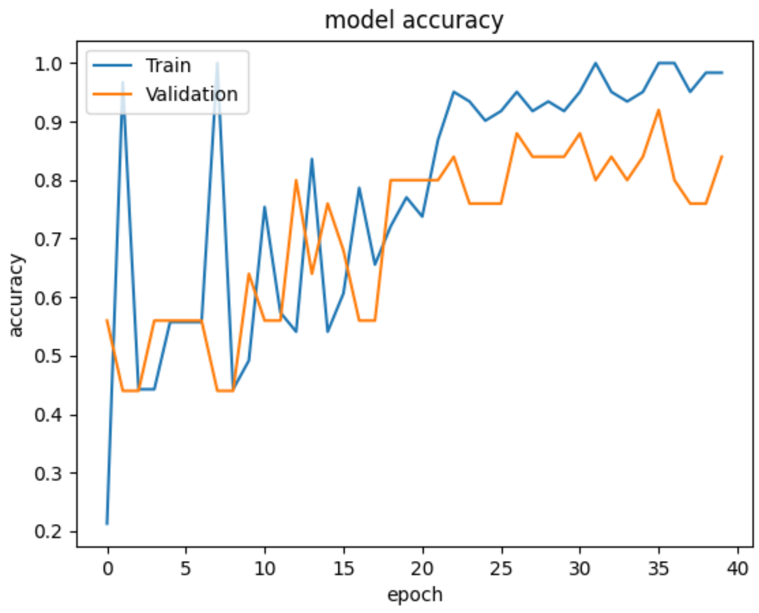


Figure 5.4 *Training and validation accuracy*

5.3.2. ALGORITHM FOR CLASSIFICATION – XCEPTION

This model uses Xception to classify the type of diabetic retinopathy. The "Extreme Version of Inception" is what Google refers to as Xception. Xception is a convolutional neural network and is 71 layers deep. On the ImageNet dataset, Inception V3 is only marginally outperformed by Xception, which performs vastly better on a larger dataset of 17,000 classes for image classification. The fact that it has the same number of model parameters as Inception and is thus computationally more efficient is most important. The entry flow is the first step in this process, after which the middle flow—which is repeated eight times—and exit flow are applied to the data.

(A) ORIGINAL DEPTHWISE SEPARABLE CONVOLUTION IN XCEPTION

The depth-wise convolution followed by the pointwise convolution is the original depth-wise separable convolution.

1. The channel-wise nxn spatial convolution is the depth-wise convolution. Given the five channels in the figure above, then there would be five times as many spatial convolutions.

2. In order to change the dimensions, the pointwise convolution is a 1x1 convolution. There is no necessity to conduct convolution among all the channels, as we would with conventional convolution. As a result, there are fewer connections, and the model is lighter.

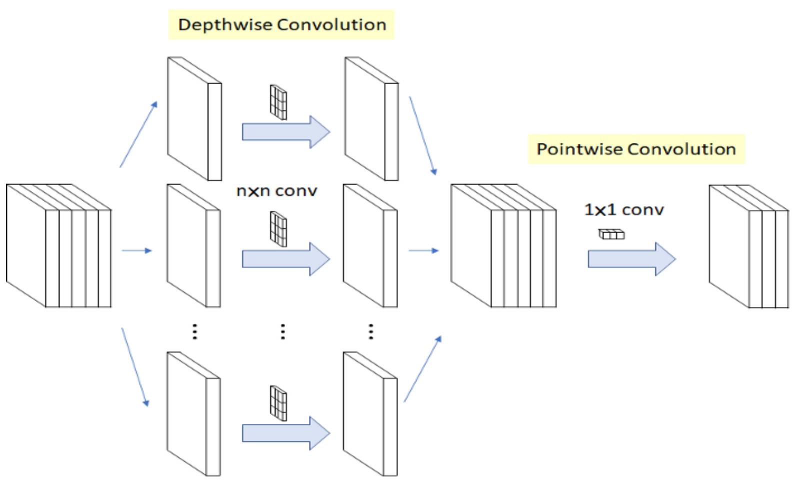


Figure 5.5 *Original Depthwise Separable Convolution in Xception*

(B) MODIFIED DEPTHWISE SEPARABLE CONVOLUTION IN XCEPTION

The pointwise convolution is followed by a depth-wise convolution, which is the modified depth-wise separable convolution. The Inception V3 module's requirement 1x1 convolutions can be performed before any nxn spatial convolutions served as the impetus for this modification. As a result, it differs slightly from the original.

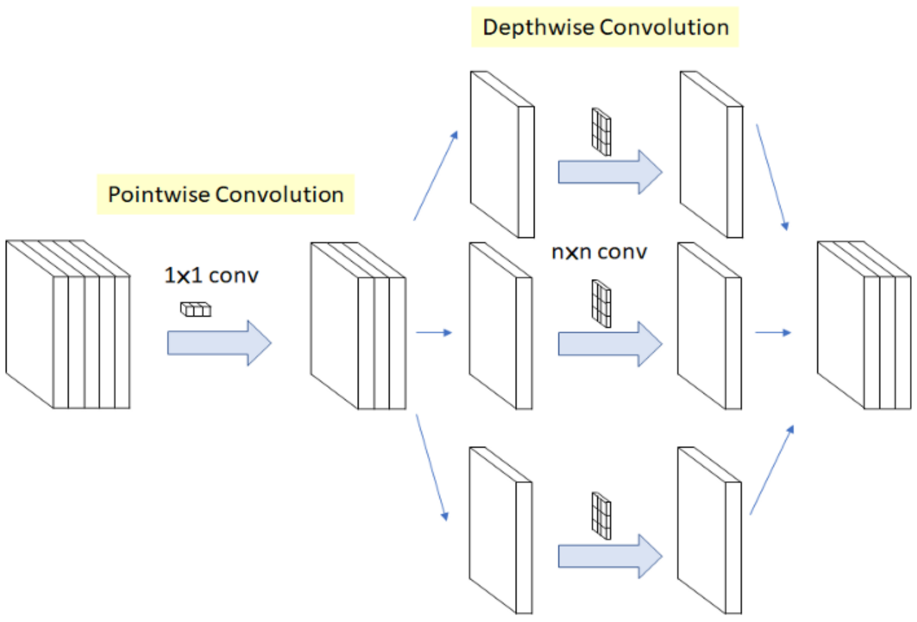


Figure 5.6 *Modified Depthwise Separable Convolution in Xception*

TWO MINOR DIFFERENCES:

1. The Presence or Absence of Non-Linearity: After the initial operation, the original Inception Module displays nonlinearity. There is no intermediary ReLU nonlinearity in Xception. The updated depth-wise separable convolution with various activation units is put to the test. The Xception without an intermediary activation has the maximum accuracy when compared to the ones employing either ReLU or ELU.

2. The order of operations: In contrast to the original depth-wise separable convolutions, which are typically implemented, the modified depth-wise separable convolution performs a 1x1 convolution first, followed by a channel-wise spatial convolution (for example, in TensorFlow). This is said to be insignificant because, when used in stacked settings, all of the chained inception modules only exhibit slight variations at their beginning and conclusion.

5.1.2. TRAINING PROCESS FOR DR CLASSIFICATION

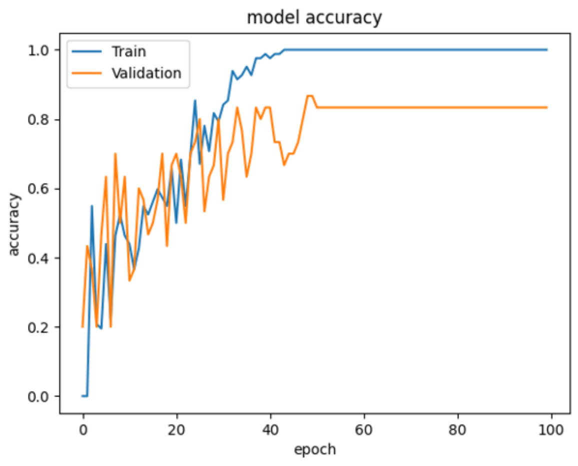


Figure 5.7 *Training and validation accuracy*

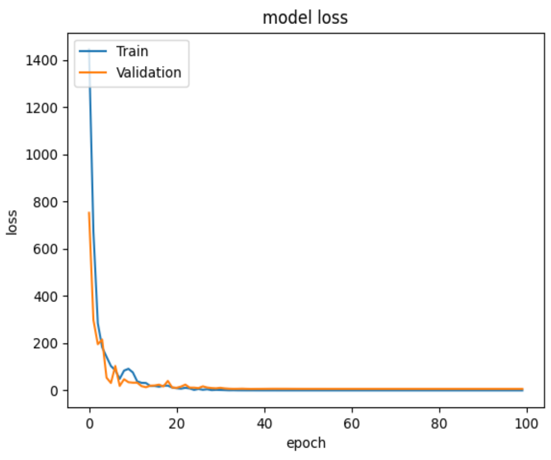


Figure 5.8 *Training and validation loss*

CHAPTER 6

SYSTEM IMPLEMENTATION

6.1 SERVER-SIDE CODING

app.py

```
# Web application for DR detection and classification
import streamlit as st
import base64
import cv2
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt
import tensorflow as tf
from keras.models import load_model
# from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Background Image

def get_base64(bin_file):
with open(bin_file, 'rb') as f:
data = f.read()
return base64.b64encode(data).decode()

def set_background(png_file):
bin_str = get_base64(png_file)
page_bg_img = '''
<style>
.stApp {
background-image: url("data:image/jpeg;base64,%s");
background-size: cover;
background-position: center;
```

```
 }
</style>

''' % bin_str
st.markdown(page_bg_img, unsafe_allow_html=True)
set_background('./images/33.jpeg')
title = '<p style="font-family:Brush Script MT; color:white; font-size: 42px;">Diabetic Retinopathy Detection and Classification</p>'
st.markdown(title, unsafe_allow_html=True)
# st.markdown('<h1 style="color:black;">Diabetic Retinopathy Detection and Classification</h1>', unsafe_allow_html=True)
upload = st.file_uploader('Insert image for classification', type=[
                    'png', 'jpg', 'jpeg'])
# c1, c2, c3 = st.columns(3)
prediction1 = []
```

# Upload Image

```
if upload is not None:
im = Image.open(upload)
img = np.asarray(im)
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
st.markdown('<h3 style="color:white;">Input Image</h3>', unsafe_allow_html=True)
#  st.subheader('<h1>Input Image</h1>')
st.image(im)
FI = cv2.medianBlur(img, 3)
if st.button('Submit'):
```

# DR detection

```
    OP1 = load_model( r'C:\Users\Computer\Desktop\project-docs\DR app\Model1.h5')
    IMG_SIZE = 224
    IM = np.array(FI)
    img_array = cv2.cvtColor(IM, cv2.COLOR_BGR2RGB)
    new_array = cv2.resize(img_array, (IMG_SIZE, IMG_SIZE))
    N = new_array.reshape(-1, IMG_SIZE, IMG_SIZE, 3)
    prediction = OP1.predict(N)
    class_labels = ['Diabetic Retinopathy', 'Normal']
    prediction1 = np.argmax(prediction, axis=-1)
    label = class_labels[prediction1[0]]
    st.markdown('<h3 style="color:white;">Detection Result :</h3>',
                unsafe_allow_html=True)
    st.write(label)
```

# DR classification

```
if prediction1 == 1:
 st.markdown('<h3 style="color:white;">Detection Result :</h3>',
unsafe_allow_html=True)
st.write('None')
elif prediction1 == 0:
IMG_SIZE = 224
IM = np.array(FI)
img_array = cv2.cvtColor(IM, cv2.COLOR_BGR2RGB)
new_array = cv2.resize(img_array, (IMG_SIZE, IMG_SIZE))
N = new_array.reshape(-1, IMG_SIZE, IMG_SIZE, 3)

OP2 = load_model( r'C:\Users\Computer\Desktop\project-docs\DR app\Model2.h5')
```

```
    prediction = OP2.predict(N)

    class_labels = ['Exudates', 'Hemorrhages', 'Microaneurysms']
    prediction = np.argmax(prediction, axis=-1)
    label = class_labels[prediction[0]]
        st.markdown('<h3 style="color:white;">Classification Result :</h3>',
unsafe_allow_html=True)
    st.write(label)
else:
    st.write()
```

app1.py

```
# Web application for DR detection and classification

from mlxtend.plotting import plot_confusion_matrix
import pandas as pd  # to read and manipulating data
from sklearn import metrics
import seaborn as sns  # for better visualization of graph with the help of Matplotlib
from sklearn.metrics import classification_report
import sklearn.metrics as metrics
from keras.preprocessing.image import ImageDataGenerator
import streamlit as st
import base64
import cv2
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt
import tensorflow as tf
```

```
from keras.models import load_model
# from tensorflow.keras.preprocessing.image import ImageDataGenerator

st.markdown('<h1 style="color:black;">Diabetic Retinopathy Detection and
Classification</h1>',
            unsafe_allow_html=True)

upload = st.file_uploader('Insert image for classification', type=[
                         'png', 'jpg', 'jpeg'])
c1, c2, c3 = st.columns(3)
prediction1 = []

# Upload Image

if upload is not None:
im = Image.open(upload)
img = np.asarray(im)
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
c1.header('Input Image')
c1.image(im)

FI = cv2.medianBlur(img, 3)

if st.button('Submit'):

# DR detection
OP1 = load_model( r'C:\Users\Computer\Desktop\project-docs\DR app\Model1.h5')
IMG_SIZE = 224
IM = np.array(FI)
```

```
    img_array = cv2.cvtColor(IM, cv2.COLOR_BGR2RGB)
    new_array = cv2.resize(img_array, (IMG_SIZE, IMG_SIZE))
    N = new_array.reshape(-1, IMG_SIZE, IMG_SIZE, 3)

    prediction = OP1.predict(N)

    class_labels = ['Diabetic Retinopathy', 'Normal']
    prediction1 = np.argmax(prediction, axis=-1)

    label = class_labels[prediction1[0]]
    c2.header('Detection Result')
    # c2.subheader('DR Detection :')
    c2.write(label)
```

# DR classification

```
if prediction1 == 1:
c3.header('Classification Result')
c3.subheader('DR Type :')
c3.write('Eye is in normal condition')
c3.write('Exit')

elif prediction1 == 0:

    IMG_SIZE = 224
    IM = np.array(FI)
    img_array = cv2.cvtColor(IM, cv2.COLOR_BGR2RGB)
    new_array = cv2.resize(img_array, (IMG_SIZE, IMG_SIZE))
    N = new_array.reshape(-1, IMG_SIZE, IMG_SIZE, 3)
```

```
OP2 = load_model( r'C:\Users\Computer\Desktop\project-docs\DR app\Model2.h5')
prediction = OP2.predict(N)

class_labels = ['Exudates', 'Hemorrhages', 'Microaneurysms']
prediction = np.argmax(prediction, axis=-1)
label = class_labels[prediction[0]]
c3.header('Classification Result')
c3.write('Eye is in abnormal condition')
c3.write(label)
else:
    st.write()
```

# Performance Measure for DR Detection

```
st.header('Performance Analysis for DR Detection')
test_dir = r'C:\Users\Computer\Desktop\project-docs\DR app\FDATASET\PA'
testdata = ImageDataGenerator()
test_data_gen = testdata.flow_from_directory(
    directory=test_dir, target_size=(224, 224), shuffle=False, class_mode='binary')
batch_size = 32
OP = load_model(r'C:\Users\Computer\Desktop\project-docs\DR app\Model1.h5')
Y_pred = OP.predict(test_data_gen, test_data_gen.samples / batch_size)
test_preds = np.argmax(Y_pred, axis=1)
test_trues = test_data_gen.classes

confusion_matrix = metrics.confusion_matrix(test_preds, test_trues)
lang = ['Diabetic Retinopathy', 'Normal']
conf_matrix_df = pd.DataFrame(confusion_matrix, columns=lang, index=lang)
plot_confusion_matrix(conf_mat=confusion_matrix,
class_names=lang,
show_absolute=False,
show_normed=True,
colorbar=True)
st.set_option('deprecation.showPyplotGlobalUse', False)
st.pyplot()
st.text('Model Report:\n    ' + classification_report(test_trues, test_preds))
```

# Performance Measure for DR Classification

```
st.header('Performance Analysis for DR Classification')
test_dir = r'C:\Users\Computer\Desktop\project-docs\DR app\SDATASET\PA'
testdata = ImageDataGenerator()
test_data_gen = testdata.flow_from_directory(directory=test_dir, target_size=(
    224, 224), shuffle=False, class_mode='categorical')
batch_size = 32
OP1 = load_model(r'C:\Users\Computer\Desktop\project-docs\DR app\Model2.h5')
Y_pred = OP1.predict(test_data_gen, test_data_gen.samples / batch_size)
test_preds = np.argmax(Y_pred, axis=1)
test_trues = test_data_gen.classes
confusion_matrix = metrics.confusion_matrix(test_preds, test_trues)
lang = ['Exudates', 'Hemorrhages', 'Microaneurysms']
conf_matrix_df = pd.DataFrame(confusion_matrix, columns=lang, index=lang)
plot_confusion_matrix(conf_mat=confusion_matrix,
class_names=lang,
show_absolute=False,
show_normed=True,
colorbar=True)
st.set_option('deprecation.showPyplotGlobalUse', False)
st.pyplot()
st.text('Model Report:\n    ' + classification_report(test_trues, test_preds))
```

<u>train1.py</u>

```
# Training Process for DR Detection
import cv2
import matplotlib.pyplot as plt
from PIL import Image
import tensorflow as tf
from keras import backend as K
from keras.models import load_model
from keras.models import save_model
from tensorflow.keras.optimizers import Adam, SGD
from tensorflow.keras.callbacks import ReduceLROnPlateau
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import numpy as np

train_dir = r'C:\Users\Computer\Desktop\project-docs\DR app\FDATASET\train'
valid_dir= r'C:\Users\Computer\Desktop\project-docs\DR app\FDATASET\validation'

IMG_SHAPE = 224
batch_size = 32

image_gen_train = ImageDataGenerator()
train_data_gen = image_gen_train.flow_from_directory(batch_size=batch_size,
directory=train_dir,
shuffle=False,
target_size=(
IMG_SHAPE, IMG_SHAPE),
class_mode='binary')

image_generator_validation = ImageDataGenerator()
val_data_gen=image_generator_validation.flow_from_directory(batch_size=batch_siz
e,
directory=valid_dir,
shuffle=False,
 target_size=(IMG_SHAPE, IMG_SHAPE),
 class_mode='binary')

pre_trained_model = tf.keras.applications.InceptionV3(
input_shape=(224, 224, 3), include_top=False, weights="imagenet")
for layer in pre_trained_model.layers:
layer.trainable = False
last_layer = pre_trained_model.get_layer('mixed10')
last_output = last_layer.output
x = tf.keras.layers.Flatten()(last_output)
x = tf.keras.layers.Dense(512, activation='relu')(x)
x = tf.keras.layers.Dense(64, activation='relu')(x)
x = tf.keras.layers.Dense(2, activation='softmax')(x)
model = tf.keras.Model(pre_trained_model.input, x)
model.compile(loss='sparse_categorical_crossentropy',
            optimizer='Adam', metrics=['acc'])

training_steps_per_epoch = np.ceil(train_data_gen.samples / batch_size)
validation_steps_per_epoch = np.ceil(val_data_gen.samples / batch_size)

history = model.fit(train_data_gen, steps_per_epoch=training_steps_per_epoch,
epochs=40,
validation_data=val_data_gen,
validation_steps=validation_steps_per_epoch,
batch_size=batch_size,
verbose=1)
print('Training Completed!')

model.save(r'C:\Users\Computer\Desktop\project-docs\DR app\Model1a.h5')
```

# summarize history for accuracy

```
history_dict = history.history
print(history_dict.keys())
acc = history_dict['acc']
val_acc = history_dict['val_acc']
loss = history_dict['loss']
val_loss = history_dict['val_loss']
fig1 = plt.figure(1)
plt.plot(acc)
plt.plot(val_acc)
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.grid(False)
plt.show()
# summarize history for loss
fig2 = plt.figure(2)
plt.plot(loss)
plt.plot(val_loss)
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.grid(False)
plt.show()
print('completed')
exit()
```

<u>train2.py</u>

# Training Process for DR Classification

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import sklearn.metrics as metrics
import tensorflow as tf
from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Flatten, Dense
from tensorflow.keras.applications import Xception
from keras.models import load_model

num_classes = 3
IMAGE_SHAPE = [224, 224]
batch_size = 32  # change for better accuracy based on your dataset
epochs = 100  # change for better accuracy based on your dataset
train_dir = r'C:\Users\Computer\Desktop\project-docs\DR app\SDATASET\train'
valid_dir= r'C:\Users\Computer\Desktop\project-docs\DR app\SDATASET\validation'
EF = Xception(input_shape=(224, 224, 3), weights="imagenet", include_top=False)
```

```
for layer in EF.layers:
layer.trainable = False
x = Flatten()(EF.output)
x = Dense(512, activation='relu')(x)
x = Dense(512, activation='relu')(x)
x = Dense(num_classes, activation='softmax')(x)
model = Model(inputs=EF.input, outputs=x)
model.compile(loss='categorical_crossentropy', optimizer='Adam', metrics=['acc'])
trdata = ImageDataGenerator()
train_data_gen = trdata.flow_from_directory(directory=train_dir, target_size=(
224, 224), shuffle=False, class_mode='categorical')
valdata = ImageDataGenerator()
val_data_gen = valdata.flow_from_directory(directory=valid_dir, target_size=(
224, 224), shuffle=False, class_mode='categorical')

training_steps_per_epoch = np.ceil(train_data_gen.samples / batch_size)
validation_steps_per_epoch = np.ceil(val_data_gen.samples / batch_size)
history = model.fit(train_data_gen, steps_per_epoch=training_steps_per_epoch,
                                                                validation_data=val_data_gen,
validation_steps=validation_steps_per_epoch, epochs=epochs, verbose=1)
print('Training Completed!')

model.save(r'C:\Users\Computer\Desktop\project-docs\DR app\Model2a.h5')

# summarize history for accuracy
history_dict = history.history
print(history_dict.keys())
acc = history_dict['acc']
val_acc = history_dict['val_acc']
```

```
loss = history_dict['loss']
val_loss = history_dict['val_loss']
fig1 = plt.figure(1)
plt.plot(acc)
plt.plot(val_acc)
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.grid(False)
plt.show()

# summarize history for loss
fig2 = plt.figure(2)
plt.plot(loss)
plt.plot(val_loss)
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.grid(False)
plt.show()
print('completed')
exit()
```

P1.py

```
# Training Process for DR Classification
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import sklearn.metrics as metrics
import tensorflow as tf
from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Flatten, Dense
from tensorflow.keras.applications import Xception
from keras.models import load_model
num_classes = 3
IMAGE_SHAPE = [224, 224]
batch_size = 32  # change for better accuracy based on your dataset
epochs = 100  # change for better accuracy based on your dataset

train_dir = r'C:\Users\Computer\Desktop\project-docs\DR app\SDATASET\train'
valid_dir= r'C:\Users\Computer\Desktop\project-docs\DR app\SDATASET\validation'

EF = Xception(input_shape=(224, 224, 3), weights="imagenet", include_top=False)
for layer in EF.layers:
    layer.trainable = False
x = Flatten()(EF.output)
x = Dense(512, activation='relu')(x)
x = Dense(512, activation='relu')(x)
x = Dense(num_classes, activation='softmax')(x)
model = Model(inputs=EF.input, outputs=x)
model.compile(loss='categorical_crossentropy',
              optimizer='Adam', metrics=['acc'])
trdata = ImageDataGenerator()
train_data_gen = trdata.flow_from_directory(directory=train_dir, target_size=(
    224, 224), shuffle=False, class_mode='categorical')
```

```
valdata = ImageDataGenerator()
val_data_gen = valdata.flow_from_directory(directory=valid_dir, target_size=(
224, 224), shuffle=False, class_mode='categorical')
training_steps_per_epoch = np.ceil(train_data_gen.samples / batch_size)
validation_steps_per_epoch = np.ceil(val_data_gen.samples / batch_size)
history = model.fit(train_data_gen, steps_per_epoch=training_steps_per_epoch,
                                                validation_data=val_data_gen,
validation_steps=validation_steps_per_epoch, epochs=epochs, verbose=1)
print('Training Completed!')

model.save(r'C:\Users\Computer\Desktop\project-docs\DR app\Model2a.h5')
# summarize history for accuracy
history_dict = history.history
print(history_dict.keys())
acc = history_dict['acc']
val_acc = history_dict['val_acc']
loss = history_dict['loss']
val_loss = history_dict['val_loss']
fig1 = plt.figure(1)
plt.plot(acc)
plt.plot(val_acc)
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.grid(False)
plt.show()
# summarize history for loss
```

```
fig2 = plt.figure(2)
plt.plot(loss)
plt.plot(val_loss)
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.grid(False)
plt.show()
print('completed')
exit()
```

P2.py

# Performance analysis for DR Classification

```
import tensorflow as tf
from keras.preprocessing.image import ImageDataGenerator
import sklearn.metrics as metrics
from sklearn.metrics import classification_report
import seaborn as sns  # for better visualization of graph with the help of Matplotlib
from sklearn import metrics
import pandas as pd  # to read and manipulating data
import numpy as np
from keras.models import load_model
import matplotlib.pyplot as plt
test_dir = r'C:\Users\Computer\Desktop\project-docs\DR app\SDATASET\PA'
testdata = ImageDataGenerator()
test_data_gen = testdata.flow_from_directory(directory=test_dir, target_size=(
    224, 224), shuffle=False, class_mode='categorical')
batch_size = 32
```

```
OP1 =load_model(r'C:\Users\Computer\Desktop\project-docs\DR app\Model2.h5')

Y_pred = OP1.predict(test_data_gen, test_data_gen.samples / batch_size)
test_preds = np.argmax(Y_pred, axis=1)
test_trues = test_data_gen.classes
# Printing class dictionary
print(test_data_gen.class_indices)
print('Classified Result:', test_preds)
print('Ture Result:', test_trues)
confusion_matrix = metrics.confusion_matrix(test_preds, test_trues)
lang = ['Exudates', 'Hemorrhages', 'Microaneurysms']
conf_matrix_df = pd.DataFrame(confusion_matrix, columns=lang, index=lang)
sns.set(font_scale=1.0)
sns.heatmap(conf_matrix_df,
            annot=True)
plt.xlabel('Classified Labels', fontsize=12)
plt.ylabel('True Labels', fontsize=12)
plt.title('Confusion matrix for DR classification')
plt.show()
print(classification_report(test_trues, test_preds))
print('completed')
exit()
```

testing.py

```
##### Testing Process ####

# Read test image
import cv2
```

```
from PIL import Image
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
from keras.models import load_model
# from tensorflow.keras.preprocessing.image import ImageDataGenerator

img = Image.open(
    r"C:\Users\Computer\Desktop\project-docs\DR app\Test Images\E1.jpg")
img = np.asarray(img)
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
# fig1 = plt.figure()
plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
plt.title('Input Image')
plt.grid(False)
plt.show()
```

# Noise removal using median filter

```
FI = cv2.medianBlur(img, 3)
# fig2 = plt.figure()
plt.imshow(cv2.cvtColor(FI, cv2.COLOR_BGR2RGB))
plt.title('Noise Removal using Median Filter')
plt.grid(False)
plt.show()
```

# Load Trained Model

```
OP1 = load_model(r'C:\Users\Computer\Desktop\project-docs\DR app\Model1.h5')
```

# DR Detection

```
IMG_SIZE = 224
IM = np.array(FI)
img_array = cv2.cvtColor(IM, cv2.COLOR_BGR2RGB)
new_array = cv2.resize(img_array, (IMG_SIZE, IMG_SIZE))
N = new_array.reshape(-1, IMG_SIZE, IMG_SIZE, 3)
print(N.shape)

prediction = OP1.predict(N)

class_labels = ['Diabetic Retinopathy', 'Normal']
prediction = np.argmax(prediction, axis=-1)
print(prediction)
print(class_labels[prediction[0]])
label = class_labels[prediction[0]]

font = cv2.FONT_HERSHEY_SIMPLEX
orgin = (550, 300)
fontScale = 10
color = (0, 255, 0)
thickness = 10
image = cv2.putText(IM, label, orgin, font,
                    fontScale, color, thickness, cv2.LINE_AA)

plt.grid(False)
plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
plt.title('DR Detection Result based on Inception V3')
plt.show()
```

# DR Classification

```
if prediction == 1:
    print('Eye condition is normal')
    print('Quit')

elif prediction == 0:
    IMG_SIZE = 224
    IM = np.array(FI)
    img_array = cv2.cvtColor(IM, cv2.COLOR_BGR2RGB)
    new_array = cv2.resize(img_array, (IMG_SIZE, IMG_SIZE))
    N = new_array.reshape(-1, IMG_SIZE, IMG_SIZE, 3)

    OP2 = load_model(r'C:\Users\Computer\Desktop\project-docs\DR app\Model2.h5')
    prediction = OP2.predict(N)

    class_labels = ['Exudates', 'Hemorrhages', 'Microaneurysms']
    prediction = np.argmax(prediction, axis=-1)
    print(prediction)
    print(class_labels[prediction[0]])
    label = class_labels[prediction[0]]

font = cv2.FONT_HERSHEY_SIMPLEX
orgin = (550, 300)
fontScale = 10
color = (0, 255, 0)
thickness = 10
 image = cv2.putText(IM, label, orgin, font, fontScale, color, thickness,
cv2.LINE_AA)
plt.grid(False)
plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
plt.title('DR Classification Result based on Xception')
plt.show()
print('completed')
exit()
```

CHAPTER 7

PERFORMANCE ANALYSIS

7.1. UNIT TESTING

Unit testing involves the design of test cases that validate the internal programme logic and ensure that the programme inputs produce valid outputs. It is also important to verify the internal code flow and the decision branches. The testing of individual software units of the application is done after the completion of an individual unit before integration. This is a structural test that relies on knowledge of its construction. Unit tests perform testing at the component level and also test specific business processes, applications, and/or system configurations. Unit tests make sure that each distinct path of a business process adheres precisely to the documented specifications and has inputs and outputs that are well-defined.

TEST RESULTS: All the test cases mentioned above passed successfully. No defects encountered.

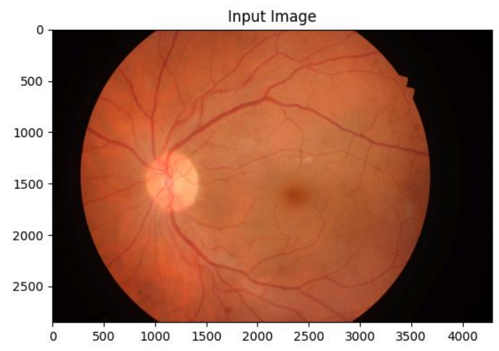
7.2. INTEGRATION TESTING

Software components that have been integrated are tested in integration tests to see if they actually operate as a single programme. Testing is event-driven and is more concerned with the basic outcome of screens or fields. Although the individual components were satisfactory, as shown by successful unit testing, integration tests show that the combination of the components is accurate and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

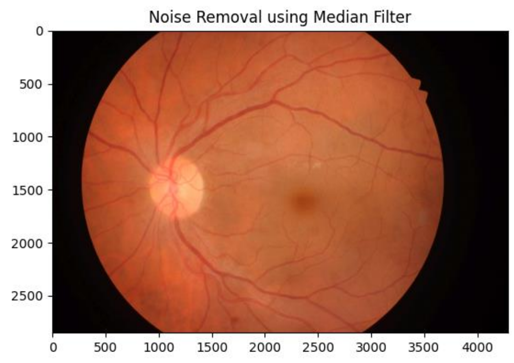
TEST RESULTS: All the test cases mentioned above passed successfully. No defects encountered.

7.2.1 TESTING PROCESS

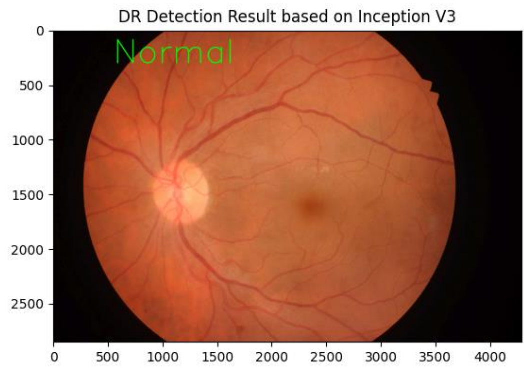
IMAGE 1: **Read Test Image**



Preprocessing



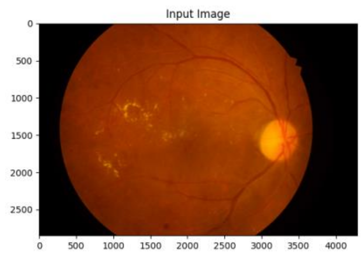
Classification



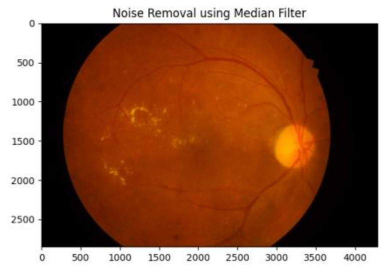
Output

```
(1, 224, 224, 3)
1/1 [==============================] - ETA: 0s 
1/1 [==============================] - 2s 2s
/step
[1]
Normal
Eye condition is normal
Quit
completed
```

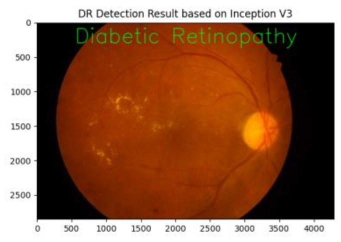
IMAGE 2: **Read Test Image**



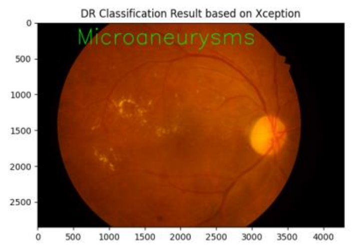
Preprocessing



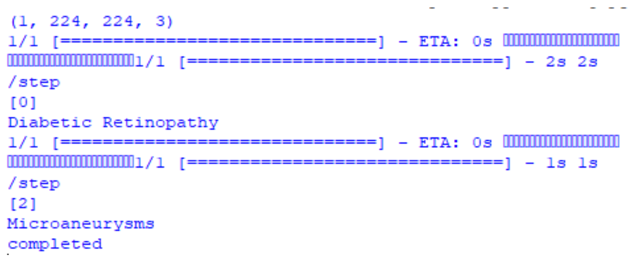
DR Detection based on Inception V3



DR Classification based on Xception



Output



```
(1, 224, 224, 3)
1/1 [==============================] - ETA: 0s ▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯1/1 [==============================] - 2s 2s
/step
[0]
Diabetic Retinopathy
1/1 [==============================] - ETA: 0s ▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯1/1 [==============================] - 1s 1s
/step
[2]
Microaneurysms
completed
```

7.3. TEST CASES & REPORTS

Action	**Input**	**Expected Output**	**Actual Output**	**Test Result**
Launch streamlit app using “streamlit run app.py”	http://localhost:8501/	DR detection and classification web app	DR detection and classification web app	Pass
Detection of diabetic retinopathy	Upload fundus image that have diabetic retinopathy	Diabetic retinopathy	Diabetic retinopathy	Pass
Classification of diabetic retinopathy	Upload fundus image that have diabetic retinopathy	Exudates	Exudates	Pass
Detection of diabetic retinopathy	Upload fundus image that don’t have diabetic retinopathy	No diabetic retinopathy	No diabetic retinopathy	Pass

Table7.1 *Test Cases*

7.4. PERFORMANCE MEASURE:

Finally, the performance of the proposed model was measured in terms of accuracy, precision, recall, and f1_score performance.

- Accuracy: It is the analysis of TP and TN against the total of test images.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \qquad (1)$$

- Precision: It is an estimation analysis of true positive to the aggregate value of true positive and false positive given in eqn.(2)

$$Precision = \frac{(TP)}{(TP+FP)} \qquad (2)$$

- Recall: It is the estimation analysis of true positive rate to the aggregate value of the true positive and false negative rate. It is given in eqn. (3).

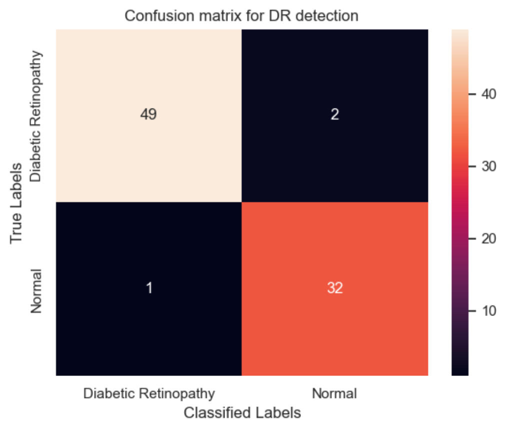
$$Recall = \frac{(TP)}{(TP+FN)} \qquad (3)$$

- F-Score: F-Measure is the harmonic mean of recall and precision. Precision and recall are given equal weight in the standard F-measure (F1).

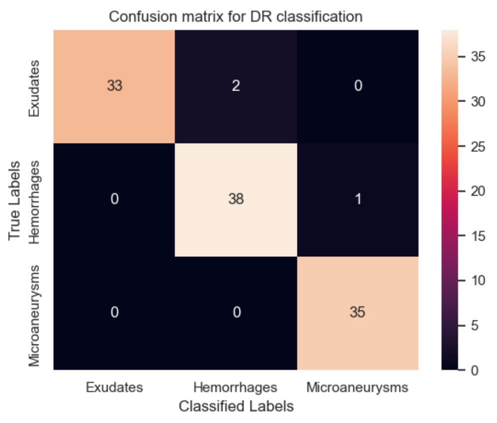
$$F_{\varepsilon} = (1+\varepsilon^{2})\frac{Recall \times Precision}{\varepsilon^{2}.\ Precision + Recall} \qquad (4)$$

```
Found 84 images belonging to 2 classes.
1/3 [=========>....................] - ETA: 14s▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯2/3 [===================>..........] - ETA: 3s ▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯3/3 [==============================] - ETA: 0s              ▯▯▯▯▯
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯3/3 [==============================]
- 12s 2s/step
{'Abnormal - Diabetic Retinopathy': 0, 'Normal': 1}
Classified Result: [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1]
Ture Result: [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1]
```

7.5 CONFUSION MATRIX



Confusion Matrix for DR Detection



Confusion Matrix for DR Classification

***Figure 7.1** Confusion Matrix*

7.6. PERFORMANCE EVALUATION:

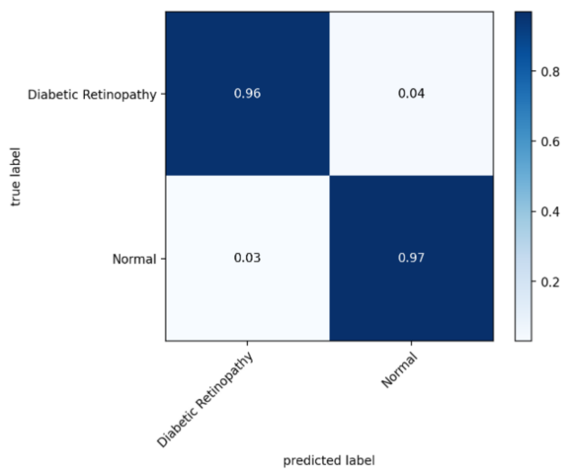


***Figure 7.2** Performance analysis for DR detection*

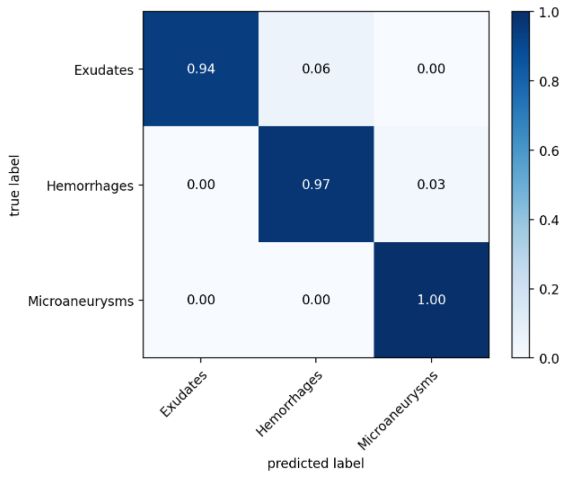
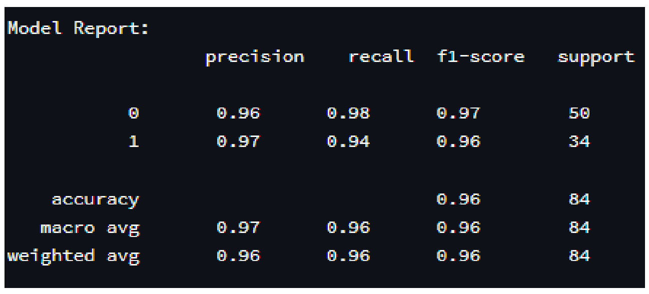


***Figure 7.3** Performance analysis for DR classification*

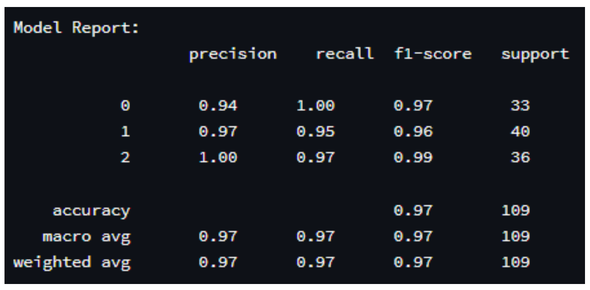


```
Model Report:
              precision    recall  f1-score   support

           0       0.96      0.98      0.97        50
           1       0.97      0.94      0.96        34

    accuracy                           0.96        84
   macro avg       0.97      0.96      0.96        84
weighted avg       0.96      0.96      0.96        84
```

***Figure 7.4** .Detection report*



```
Model Report:
              precision    recall  f1-score   support

           0       0.94      1.00      0.97        33
           1       0.97      0.95      0.96        40
           2       1.00      0.97      0.99        36

    accuracy                           0.97       109
   macro avg       0.97      0.97      0.97       109
weighted avg       0.97      0.97      0.97       109
```

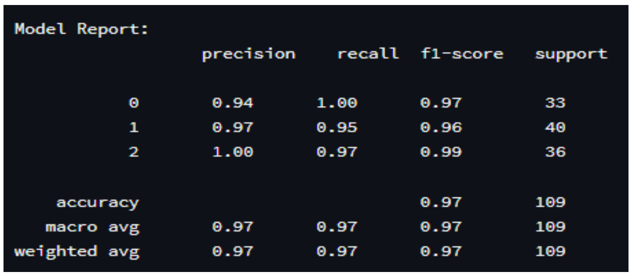
***Figure 7.5** Classification report*

The performance of the models in the identification and classification of diabetic retinopathy is shown in the confusion matrix above for the Inception V3 model and the Xception model. The Inception V3 model's accuracy is 96%, whereas the Xception model's accuracy is 97%. As a result, these models' overall accuracy is 97%, which is higher than the accuracy of all other current models. We employed 84 photos for the performance study of the Inception V3 models and 109 fundus images for the performance analysis of the Xception model.

CHAPTER 8
CONCLUSION

8.1. RESULTS & DISCUSSION

Automatic Diabetic Retinopathy grading has recently been intensively investigated in the research community, especially with advances in deep learning algorithms. Diabetic retinopathy is a diabetes-related complication that causes visual impairment. It is caused by damaging the vessels of the retina, which consists of very thin tissues. In its mild or moderate stage, Diabetic Retinopathy does not show significant symptoms. However, at a severe stage, it can lead to permanent blindness. For detection, we trained the Inception V3 model to check whether a person has diabetic retinopathy or not based on the input image (fundus image) given by that person. For classification, we trained the Xception model to determine what type of diabetic retinopathy a patient has based on the fundus image given by the patient. With our proposed model, we have achieved 97% accuracy.



```
Model Report:
              precision    recall  f1-score   support

           0       0.94      1.00      0.97        33
           1       0.97      0.95      0.96        40
           2       1.00      0.97      0.99        36

    accuracy                           0.97       109
   macro avg       0.97      0.97      0.97       109
weighted avg       0.97      0.97      0.97       109
```

Figure 8.1 *Model Report*

8.2. CONCLUSION AND FUTURE ENHANCEMENTS

In this research, we suggested a model relying on these Inception V3 and Xception architectures for the identification and classification of retinopathy.On the basis of hyper parameters, the suggested network models were enhanced for better categorization. The results of the studies show that the CNN model we've suggested can correctly categorise distinct Diabetic Retinopathy kinds. Since artificial intelligence technology is advancing, a potential future extension of this work entails the development of a smartphone app that would take eye images taken with the device's camera and process them to determine whether or not Diabetic Retinopathy has been experienced by the user and, if so, classify the types of Diabetic Retinopathy. By implementing this future model, people get to save their time from visiting the ophthalmological hospitals and keep the ophthalmological hospitals alive in remote areas through this application.

APPENDICES

A.1. SAMPLE SCREENSHOTS

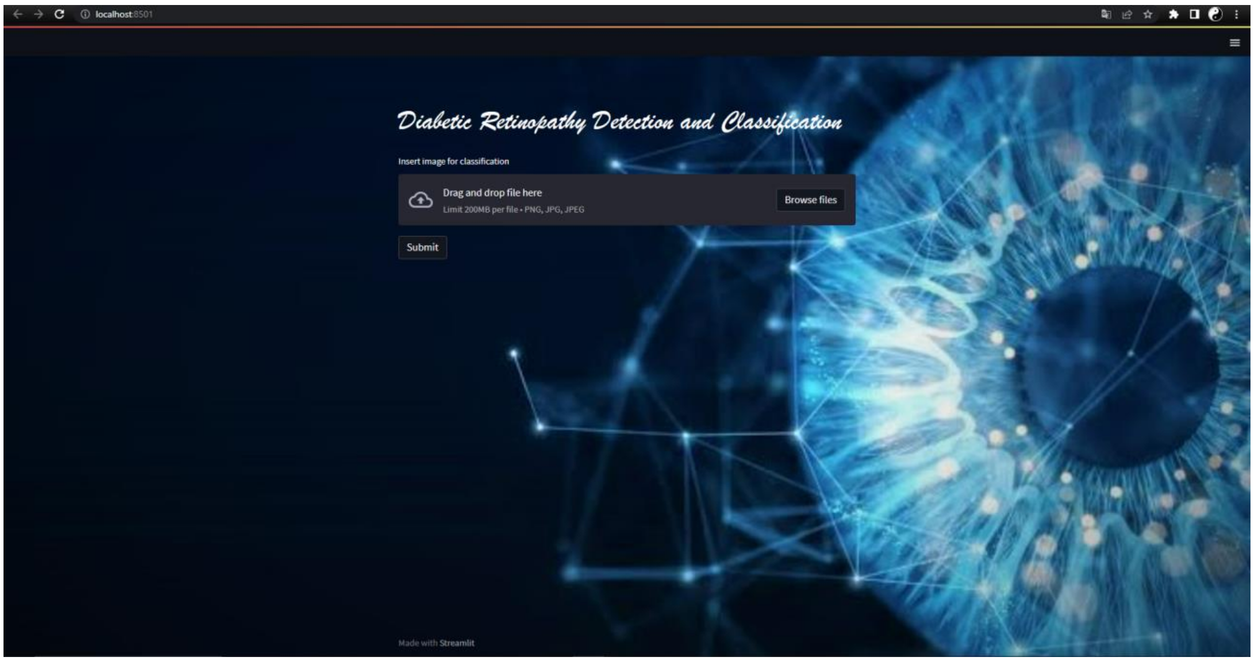

localhost:8501
Diabetic Retinopathy Detection and Classification
Insert image for classification
Drag and drop file here
Limit 200MB per file • PNG, JPG, JPEG
Browse files
Submit
Made with Streamlit

Figure A.1 *Sample Screenshot 1*

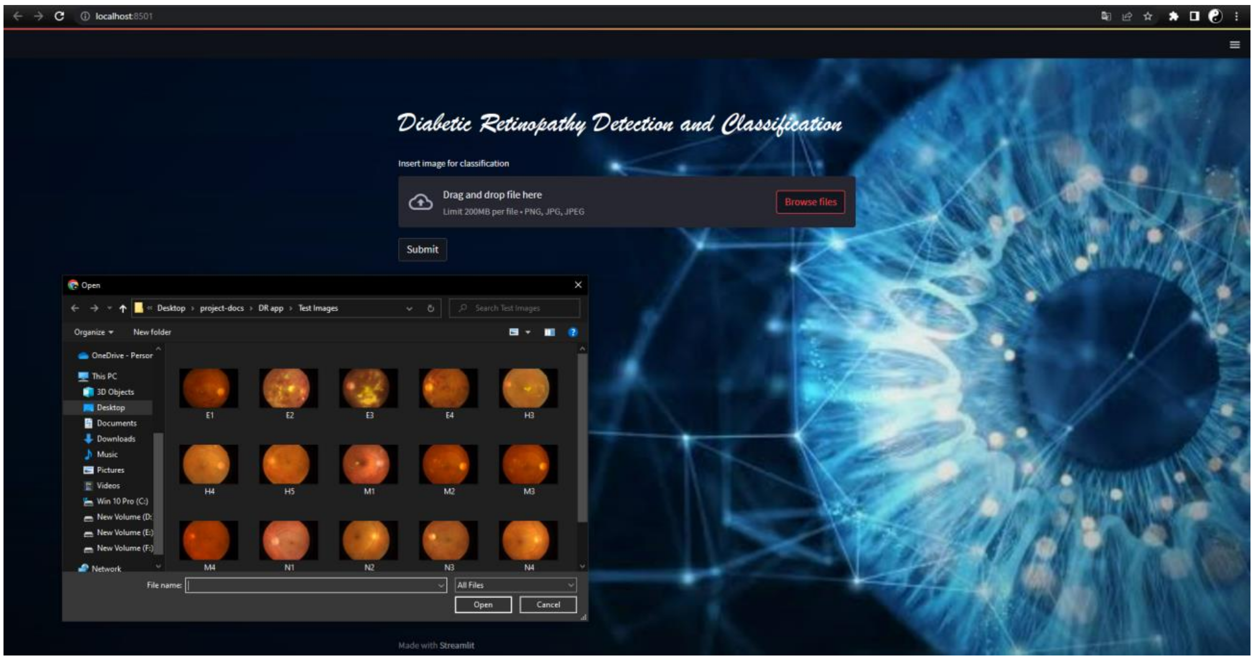


Figure A.2 *Sample Screenshot 2*

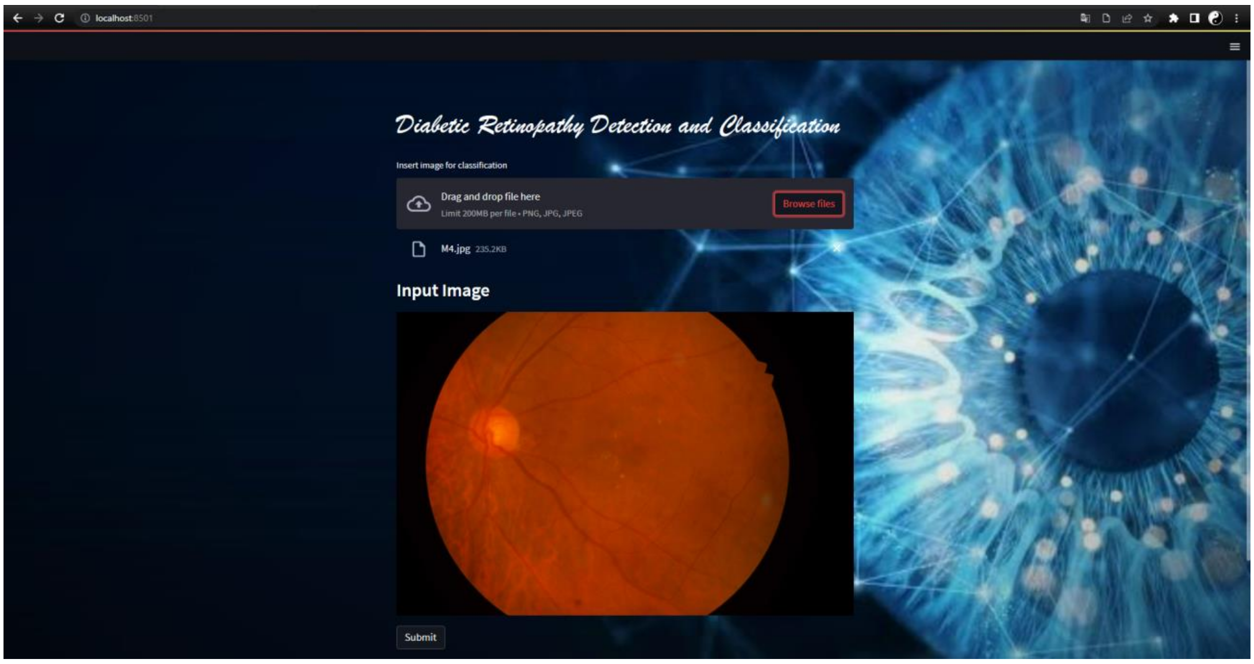

localhost:8501
Diabetic Retinopathy Detection and Classification
Insert image for classification
Drag and drop file here
Limit 200MB per file • PNG, JPG, JPEG
Browse files
M4.jpg 235.2KB
Input Image
Submit

Figure A.3 *Sample Screenshot 3*

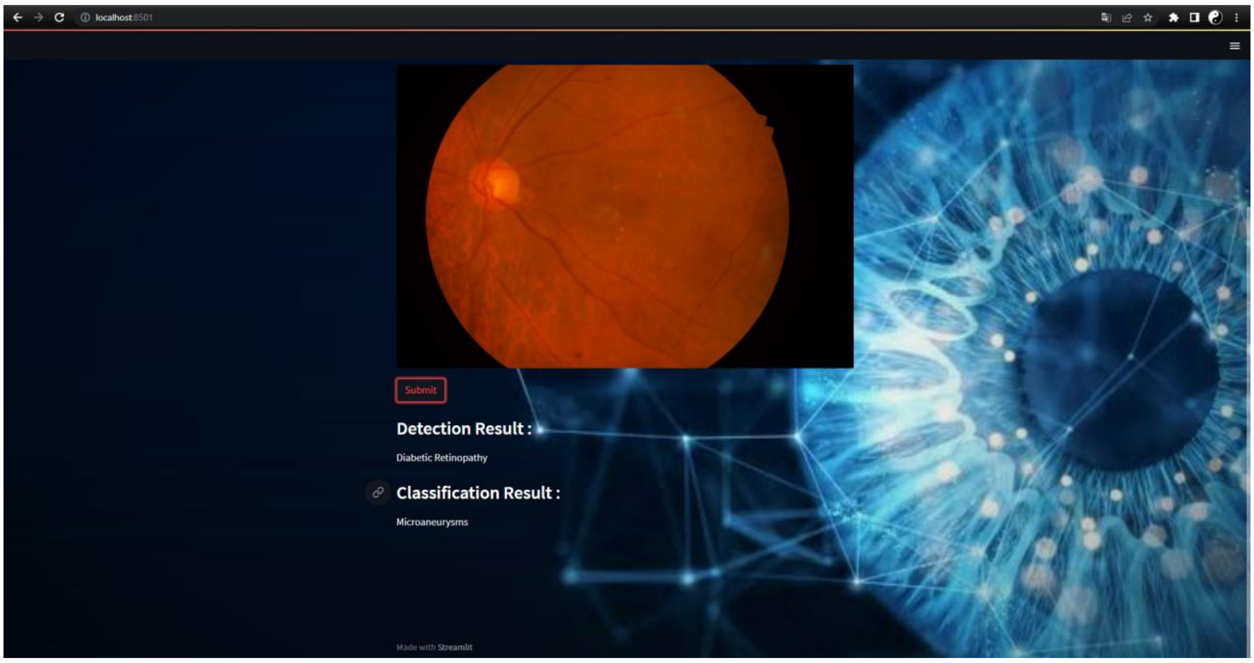


Figure A.4 *Sample Screenshot 4*

A.2. PUBLICATIONS

A.3. PLAGIARISM

REFERENCES

[1] Kaushik, H., Singh, D., Kaur, M., Alshazly, H., Zaguia, A., & Hamam, H. (2021). Diabetic retinopathy diagnosis from fundus images using stacked generalization of deep models. IEEE Access, 9, 108276-108292.

[2] Farag, M. M., Fouad, M., & Abdel-Hamid, A. T. (2022). Automatic severity classification of diabetic retinopathy based on denseNet and convolutional block attention module. IEEE Access, 10, 38299- 38308.

[3] Gao, Z., Li, J., Guo, J., Chen, Y., Yi, Z., & Zhong, J. (2018). Diagnosis of diabetic retinopathy using deep neural networks. IEEE Access, 7, 3360- 3370.

[4] Mustafa, H., Ali, S. F., Bilal, M., & Hanif, M. S. (2022). Multi-stream deep neural network for diabetic retinopathy severity classification under a boosting framework. IEEE Access, 10, 113172- 113183.

[5] Khan, Z., Khan, F. G., Khan, A., Rehman, Z. U., Shah, S., Qummar, S., ... & Pack, S. (2021). Diabetic retinopathy detection using VGG-NIN a deep learning architecture. IEEE Access, 9, 61408-61416.

[6] Saeed, F., Hussain, M., & Aboalsamh, H. A. (2021). Automatic diabetic retinopathy diagnosis using adaptive finetuned convolutional neural network. IEEE Access, 9, 41344-41359.

[7] Zeng, X., Chen, H., Luo, Y., & Ye, W. (2019). Automated diabetic retinopathy detection based on binocular siamese-like convolutional neural network. IEEE access, 7, 30744-30753.

[8] Liu, T., Chen, Y., Shen, H., Zhou, R., Zhang, M., Liu, T., & Liu, J. (2021). A novel diabetic retinopathy detection approach based on deep symmetric convolutional neural network. IEEE Access, 9, 160552-160558.

[9] Atlas, D. (2015). International diabetes federation. IDF diabetes atlas. Brussels: international diabetes federation.

[10] Das, A. (2016). Diabetic retinopathy: battling the global epidemic. Investigative ophthalmology & visual science, 57(15), 6669-6682.

[11] Rahim, S. S., Palade, V., Jayne, C., Holzinger, A., & Shuttleworth, J. (2015). Detection of diabetic retinopathy and maculopathy in eye fundus images using fuzzy image processing. In Brain Informatics and Health: 8th International Conference, BIH 2015, London, UK, August 30-September 2, 2015. Proceedings 8 (pp. 379-388). Springer International Publishing.

[12] Walter, T., Klein, J. C., Massin, P., & Erginay, A. (2002). A contribution of image processing to the diagnosis of diabetic retinopathy-detection of exudates in color fundus images of the human retina. IEEE transactions on medical imaging, 21(10), 1236- 1243.

[13] Subhashini, R., Nithin, T. N. R., & Koushik, U. M. S. (2019). Diabetic retinopathy detection using image processing (gui). International Journal of Recent Technology and Engineering, 8(2).

[14] Yun, W. L., Acharya, U. R., Venkatesh, Y. V., Chee, C., Min, L. C., & Ng, E. Y. K. (2008). Identification of different stages of diabetic retinopathy using retinal optical images. Information sciences, 178(1), 106-121.

[15] Li, X., Pang, T., Xiong, B., Liu, W., Liang, P., & Wang, T. (2017, October). Convolutional neural networks based transfer learning for diabetic retinopathy fundus image classification. In 2017 10th international congress on image and signal processing, biomedical engineering and informatics (CISP-BMEI) (pp. 1-11). IEEE.

[16] Kwasigroch, A., Jarzembinski, B., & Grochowski, M. (2018, May). Deep CNN based decision support system for detection and assessing the stage of diabetic retinopathy. In 2018 International Interdisciplinary PhD Workshop (IIPhDW) (pp. 111-116). IEEE.

[17] Masood, S., Luthra, T., Sundriyal, H., & Ahmed, M. (2017, May). Identification of diabetic retinopathy in eye images using transfer learning. In 2017 international conference on computing, communication and automation (ICCCA) (pp. 1183- 1187). IEEE.

[18] Xu, K., Feng, D., & Mi, H. (2017). Deep convolutional neural network-based early automated detection of diabetic retinopathy using fundus image. Molecules, 22(12), 2054.

[19] Rahim, S. S., Palade, V., Almakky, I., & Holzinger, A. (2019). Detection of diabetic retinopathy and maculopathy in eye fundus images using deep learning and image augmentation. In Machine Learning and Knowledge Extraction: Third IFIP TC 5, TC 12, WG 8.4, WG 8.9, WG 12.9 International Cross-Domain Conference, CD-MAKE 2019, Canterbury, UK, August 26–29, 2019, Proceedings 3 (pp. 114-127). Springer International Publishing.

[20] Doshi, D., Shenoy, A., Sidhpura, D., & Gharpure, P. (2016, December). Diabetic retinopathy detection using deep convolutional neural networks. In 2016 international conference on computing, analytics and security trends (CAST) (pp. 261-266). IEEE.

[21] Pratt, H., Coenen, F., Broadbent, D. M., Harding, S. P., & Zheng, Y. (2016). Convolutional neural networks for diabetic retinopathy. Procedia computer science, 90, 200-205.

[22] Sinthanayothin, C., Boyce, J. F., Williamson, T. H., Cook, H. L., Mensah, E., Lal, S., & Usher, D. (2002). Automated detection of diabetic retinopathy on digital fundus images. Diabetic medicine, 19(2), 105-112.

[23] Pal, R., Poray, J., & Sen, M. (2017, May). Application of machine learning algorithms on diabetic retinopathy. In 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT) (pp. 2046-2051). IEEE.

[24] Arcadu, F., Benmansour, F., Maunz, A., Willis, J., Haskova, Z., & Prunotto, M. (2019). Deep learning algorithm predicts diabetic retinopathy progression in individual patients. Npj Digital Medicine, 2 (1).

[25] Xu, L., & Luo, S. (2010). A novel method for blood vessel detection from retinal images. Biomedical engineering online, 9, 1-10.

[26] Tymchenko, B., Marchenko, P., & Spodarets, D. (2020). Deep learning approach to diabetic retinopathy detection. arXiv preprint arXiv:2003.02261.

[27] "Preprocessing in diabetic retinopathy," accessed: 2020- 08-25. [Online]. Available: https://www.kaggle.com/ratthachat/aptoseyepreprocessing-in-diabetic-retinopathy .

[28] Tan, M., & Le, Q. (2019, May). Efficientnet: Rethinking model scaling for

convolutional neural networks.

[29] Stitt, A. W., Curtis, T. M., Chen, M., Medina, R. J., McKay, G. J., Jenkins, A., ... & Lois, N. (2016). The progress in understanding and treatment of diabetic retinopathy. Progress in retinal and eye research, 51, 156-186.

[30] Rakhlin, A. (2017). Diabetic Retinopathy detection through integration of Deep Learning classification framework. BioRxiv, 225508.

[31] Lam, C., Yi, D., Guo, M., & Lindsey, T. (2018). Automated detection of diabetic retinopathy using deep learning. AMIA summits on translational science proceedings, 2018, 147.

[32] Chetoui, M., & Akhloufi, M. A. (2020). Explainable end-to-end deep learning for diabetic retinopathy detection across multiple datasets. Journal of Medical Imaging, 7(4), 044503- 044503.

[33] Kauppi, T., Kalesnykiene, V., Kamarainen, J. K., Lensu, L., Sorri, I., Raninen, A., ... & Pietilä, J. (2007, September). The diaretdb1 diabetic retinopathy database and evaluation protocol. In BMVC (Vol. 1, No. 1, p. 10).

[34] Li, T., Gao, Y., Wang, K., Guo, S., Liu, H., & Kang, H. (2019). Diagnostic assessment of deep learning algorithms for diabetic retinopathy screening. Information Sciences, 501, 511-522.

[35] Porwal, P., Pachade, S., Kamble, R., Kokare, M., Deshmukh, G., Sahasrabuddhe, V., & Meriaudeau, F. (2018). Indian diabetic retinopathy image dataset (IDRiD): a database for diabetic retinopathy screening research. Data, 3(3), 25.