A project report on

Custom Model Diabetic Retinopathy Detection

Submitted in partial fulfillment for the award of the degree of

Bachelor of Technology in Computer Science and Engineering

by

SANTHOSHKUMAR G (20BCE1184)

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CHENNAI

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

April,2024

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CHENNAI

DECLARATION

I hereby declare that the thesis entitled "CUSTOM MODEL DIABETIC RETINOPATHY DETECTION" submitted by me, for the award of the degree of Bachelor of Technology in Computer Science and Engineering, Vellore Institute of Technology, Chennai is a record of bonafide work carried out by me under the supervision of Shyamala L.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Chennai

Date: Signature of the Candidate

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CHENNAI

School of Computer Science and Engineering

CERTIFICATE

This is to certify that the report entitled **"Custom Model Diabetic Retinopathy Detection"** is prepared and submitted by Santhoshkumar G**(20BCE1184)** to Vellore Institute of Technology, Chennai, in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology in Computer Science and Engineering programme** is a bonafide record carried out under my guidance. The project fulfills the requirements as per the regulations of this University and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma and the same is certified.

Signature of the Guide:

Name: Dr./Prof.

Date:

Signature of the Internal Examiner

Name:

Date:

Signature of the External Examiner

Name:

Date:

Approved by the Head of Department,
B.Tech. CSE

Name: Dr. Nithyanandam P

Date:

(Seal of SCOPE

ABSTRACT

Diabetic retinopathy (DR) is a prevalent and debilitating condition that affects millions of people worldwide. It is an eye complication that can occur in individuals with diabetes, resulting from the impairment of blood vessels located in the retina, which is the light-sensitive layer positioned at the back of the eye. Early detection and treatment of DR are crucial to prevent vision loss and improve patient outcomes. However, the traditional method of DR detection using manual inspection of fundus photographs by ophthalmologists is time-consuming and subjective. Recent advances in computer vision and machine learning have led to the development of automated algorithms for DR detection using retinal images. However, most existing algorithms rely on a single model, which can limit their accuracy and applicability in clinical settings.

Here, a deep learning-based algorithm for DR detection that uses only fundus photographs is proposed. The proposed algorithm trains and tests two deep learning models - Convolutional Neural Network (CNN) and ResNet to classify the severity of DR. A dataset of 3662 fundus retinal images from patients with DR and No DR is used to evaluate the performance of the proposed algorithm. The images were divided into five categories based on DR severity: No DR, mild, moderate, severe and proliferate. The dataset is analyzed using the stated Deep Learning Algorithms to determine the optimum performance in terms of accuracy, precision, recall, true positive, true negative, false positive and false negative. The best algorithm is chosen after each one's overall performance has been assessed.

Experimental results show that the best algorithm achieves an accuracy of 96% in DR detection, which is higher than the accuracy achieved by single-modal algorithms. Furthermore, the algorithm is robust to noise and variability in the images, indicating its potential for clinical use.

ACKNOWLEDGEMENT

It is my pleasure to express with deep sense of gratitude to Shyamala L, Associate Professor Senior, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, for her constant guidance, continual encouragement, understanding; more than all, she taught me patience in my endeavor. My association with her is not confined to academics only, but it is a great opportunity on my part of work with an intellectual and expert in the field of AI & ML.

It is with gratitude that I would like to extend my thanks to the visionary leader Dr. G. Viswanathan our Honorable Chancellor, Mr. Sankar Viswanathan, Dr. Sekar Viswanathan, Dr. G V Selvam Vice Presidents, Dr. Sandhya Pentareddy, Executive Director, Ms. Kadhambari S. Viswanathan, Assistant Vice-President, Dr. V. S. Kanchana Bhaaskaran Vice-Chancellor i/c & Pro-Vice Chancellor, VIT Chennai and Dr. P. K. Manoharan, Additional Registrar for providing an exceptional working environment and inspiring all of us during the tenure of the course.

Special mention to Dr. Ganesan R, Dean, Dr. Parvathi R, Associate Dean Academics, Dr. Geetha S, Associate Dean Research, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai for spending their valuable time and efforts in sharing their knowledge and for helping us in every aspect.

In jubilant state, I express ingeniously my whole-hearted thanks to Dr. Nithyanandam P, Head of the Department, B.Tech. CSE and the Project Coordinators for their valuable support and encouragement to take up and complete the thesis.

My sincere thanks to all the faculties and staffs at Vellore Institute of Technology, Chennai who helped me acquire the requisite knowledge. I would like to thank my parents for their support. It is indeed a pleasure to thank my friends who encouraged me to take up and complete this task.

Place: Chennai

Date: Santhoshkumar G

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE NO
	ABSTRACT	**i**
	LIST OF FIGURES	**iv**
	LIST OF ABBREVIATIONS	**v**
1.	**INTRODUCTION** 1.1INTRODUCTION 1.2DIABETIC RETINOPATHY 1.3TYPES 1.4CAUSES OF DIABETIC RETINOPATHY 1.5SYMPTOMS OF DIABETIC RETINOPATHY 1.6DETECTION OF DIABETIC RETINOPATHY	**1** **1**
2.	**LITERATURE SURVEY**	**2**
3.	**SYSTEM ANALYSIS** 3.1 EXISTING SYSTEM 3.2 PROPOSED SYSTEM 3.3 REQUIREMENT SPECIFICATION 3.4 LANGUAGE USED–PYTHON	**7** **7** **7** **8** **8**
4.	**SYSTEM DESIGN** 4.1 SYSTEM ARCHITECTURE 4.2 DATA FLOW DIAGRAM 4.3 USE-CASE DIAGRAM 4.4 ACTIVITY DIAGRAM 4.5 SEQUENCE DIAGRAM 4.6 CLASS DIAGRAM 4.7 ER DIAGRAM	**10** **10** **10** **13** **15** **16** **16** **19**
5.	**MODULE DESCRIPTION** 5.1 MODULE 1 5.2 MODULE 2 5.3 MODULE 3	**20** **20** **22** **23**
6.	**TESTING** 6.1 TEST CASES	**27** **27**
7.	**CONCLUSION** 7.1 CONCLUSION 7.2 FUTURE ENHANCEMENT	**28** **28** **28**
8.	**APPENDIX 1** 8.1 CODING	**28**
9.	**REFERENCES**	**48**

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
4.1	SYSTEM ARCHITECTURE	**10**
4.2	DATA FLOW DIAGRAM 0	**12**
4.3	DATA FLOW DIAGRAM 1	**13**
4.4	USE-CASE DIAGRAM	**14**
4.5	ACTIVITY DIAGRAM	**15**
4.6	SEQUENCE DIAGRAM	**16**
4.7	CLASS DIAGRAM	**17**
4.8	ER DIAGRAM	**19**

LIST OF ACRONYMS AND ABBREVIATIONS

DR - Diabetic Retinopathy
CNN - Convolutional Neural Network
ResNet - Residual Neural Network
NPDR - Non-proliferative diabetic retinopathy
PDR - Proliferative diabetic retinopathy

CHAPTER 1

INTRODUCTION

1.1 Introduction

The World Health Organization (WHO) states that diabetic retinopathy is the primary cause of blindness for individuals aged 20-64 years who are still working worldwide. In 2017, approximately 4.2 million individuals were blind due to diabetic retinopathy, while another 98 million experienced moderate to severe visual impairment. A study published in the Indian Journal of Endocrinology and Metabolism in 2016 revealed that diabetic retinopathy's occurrence in India is approximately 21.7%. Furthermore, the study also found that the prevalence of diabetic retinopathy increases with the duration of diabetes and that individuals with poor glycemic control are more prone to develop diabetic retinopathy. Another study published in the Indian Journal of Ophthalmology in 2019 reported that diabetic retinopathy is the top reason for blindness in India, accounting for almost 6% of all blindness cases. The study also found that the incidence of diabetic retinopathy is higher in urban regions than in rural areas, and that there is a lack of awareness and screening programs for the disease in several parts of the country.

Among working-age adults, Diabetic retinopathy is a leading contributor to vision loss and impaired eyesight. Detecting and treating diabetic retinopathy at an early stage is crucial to avoid vision loss. Different automated techniques have been suggested for diabetic retinopathy detection using fundus images, but they have some limitations regarding accuracy and computational efficiency. In recent times, there has been an increasing focus on the use of multi-modal algorithms for detecting diabetic retinopathy, which incorporate data from various imaging modalities such as fundus images and visual field data.

1.2 Diabetic Retinopathy

Diabetic retinopathy is an eye-related complication that arises from diabetes. It results from the damage of small blood vessels in the retina, which detects light and transmits signals to the brain. The damage is caused by high blood sugar levels, which can lead to fluid or blood leakage from the blood vessels or their closure, causing inadequate circulation and retina damage. This damage can result in visual issues or blindness, and diabetic retinopathy is one

of the major causes of blindness among adults. However, with effective diabetes management and frequent eye check-ups, diabetic retinopathy can be prevented or controlled.

1.3 Types

There are two primary forms of diabetic retinopathy:

Non-proliferative diabetic retinopathy (NPDR): This is the initial stage of diabetic retinopathy that involves minor damages to the blood vessels in the retina. In this stage, the blood vessels weaken and start to leak, resulting in small amounts of fluid and blood seeping into the retina. This can lead to inflammation in the macula, which is responsible for sharp, central vision. With the progression of NPDR, the blockage of more and more blood vessels in the retina can deprive the retina of essential oxygen and nutrients.

Proliferative diabetic retinopathy (PDR): This is the advanced stage of diabetic retinopathy, where new blood vessels start to grow in the retina. These new blood vessels are delicate and can easily break, resulting in the leakage of blood and fluid into the vitreous, transparent gel-like substance in the eye. This condition can cause the vitreous to become hazy and contribute to vision loss. Additionally, the growth of new blood vessels can stimulate the formation of scar tissue, which can pull on the retina and lead to retinal detachment. PDR is more severe than NPDR and can result in greater visual impairment.

1.4 Causes of Diabetic Retinopathy

The damage to the blood vessels in the retina due to high blood sugar levels is the root cause of diabetic retinopathy. Although the exact mechanism behind how high blood sugar levels damage blood vessels is not fully comprehended, it is believed that chronically elevated blood sugar levels can result in alterations in the blood vessels that increase their susceptibility to leakage, blockage, or abnormal growth. There are additional risk factors that can contribute to the onset and advancement of diabetic retinopathy, including hypertension, high cholesterol, smoking, and having diabetes for an extended period.

1.5 Symptoms of Diabetic Retinopathy

In its early stages, diabetic retinopathy may not show any noticeable symptoms. However, as the condition progresses, symptoms may include:

1. Blurred or fluctuating vision

2. Impaired color vision
3. Dark or empty areas in the field of vision
4. Floaters or spots in the vision
5. Difficulty seeing at night
6. Sudden loss of vision

1.6 Detection of Diabetic Retinopathy

Various techniques have been devised to identify diabetic retinopathy, including manual inspection by eye specialists and computer-based algorithms for image analysis. Nevertheless, these approaches have their limitations, such as being prone to subjectivity and requiring high costs for manual inspection, and lacking accuracy and sensitivity for automated analysis.

Researchers have endeavored to overcome these obstacles by exploring the potential of multi-modal algorithms that amalgamate various imaging modalities and employ machine-learning techniques to enhance the accuracy and sensitivity of diabetic retinopathy detection. These algorithms can scrutinize images from diverse sources, such as fundus photographs and fluorescence angiography, in order to recognize and categorize different phases of diabetic retinopathy.

Diabetic retinopathy can be detected using deep learning technology that leverages computer vision methods to examine retinal images and detect signs of the condition. Deep learning algorithms can be trained on extensive sets of retinal images to recognize patterns and characteristics that signify diabetic retinopathy. By doing so, these algorithms can analyze new retinal images and classify them as normal or potentially indicative of the disease.

The process of manually interpreting images for diabetic retinopathy diagnosis can be time-consuming and the accuracy of the diagnosis may vary between different observers. To overcome this challenge, machine learning and deep learning algorithms have been developed to automate the detection of diabetic retinopathy.

The integration of multi-modal algorithms has emerged as a promising approach in diabetic retinopathy detection, with several studies reporting high accuracy and sensitivity rates. By leveraging information from diverse imaging techniques, these algorithms can enhance the

early identification and treatment of diabetic retinopathy, potentially reducing the risk of vision impairment and blindness among diabetic patients.

This article aims to provide an overview of the current state of the art in diabetic retinopathy detection using multi-modal algorithms. The article first discusses the different imaging modalities used for diabetic retinopathy detection and their advantages and limitations. It then presents a review of recent studies that have proposed multi-modal algorithms for diabetic retinopathy detection and compares their performance to traditional single-modal approaches. Finally, the article identifies the key challenges and future directions in this field. The use of multi-modal algorithms has the potential to significantly improve the accuracy and efficiency of diabetic retinopathy detection, thereby enabling early intervention and prevention of vision loss.

The motivation behind research is to address the need for early and accurate detection of diabetic retinopathy, a leading cause of blindness in working-age adults. Traditional methods of detecting diabetic retinopathy involve manual examination of retinal images by ophthalmologists, which can be time-consuming, expensive, and subjective. Therefore, there is a growing need for automated and reliable methods for the early detection of diabetic retinopathy. The article proposes a multi-modal algorithm that combines retinal images and clinical data to improve the accuracy and efficiency of diabetic retinopathy detection. The ultimate goal of this research is to improve the quality of life for diabetic patients by enabling early detection and treatment of diabetic retinopathy.

CHAPTER 2

LITERATURE REVIEWS

1. "Deep learning-based algorithm for diabetic retinopathy detection using fundus photographs":

This paper proposes a deep learning-based algorithm for detecting diabetic retinopathy (DR) using fundus photographs. DR is a prevalent and debilitating condition that affects millions of people worldwide. The algorithm uses five models, including Convolutional Neural Network (CNN), ResNet, GoogleNet, InceptionV3, and VGG16, to classify the severity of DR based on fundus photographs. The dataset used to evaluate the algorithm consists of 3662 fundus retinal images from patients with DR and no DR, which were divided into five categories based on DR severity. The experimental results showed that the proposed algorithm achieved an accuracy of 97% in DR detection, which is higher than the accuracy achieved by single-modal algorithms. Moreover, the algorithm's performance was robust to noise and variability in the images, indicating its potential for clinical use.

2. "Automated grading of diabetic retinopathy using deep learning algorithms":

This paper proposes an automated system for grading diabetic retinopathy using deep learning algorithms. The system uses Inception-v3 and DenseNet to classify retinal images into one of five categories based on the severity of DR. The dataset used to evaluate the system consists of 140,000 fundus images from 26,126 patients. The system achieved an accuracy of 95.3% in DR grading, demonstrating its potential for clinical use. The system is capable of processing a large number of images in a short period, which can improve patient outcomes by enabling early detection and treatment of DR.

3. Automatic detection of diabetic retinopathy using convolutional neural network":

Diabetic retinopathy (DR) is a common complication of diabetes and a leading cause of blindness. Early detection and treatment of DR are crucial to prevent vision

loss and improve patient outcomes. In this paper, the authors propose an automatic detection system for DR using a convolutional neural network (CNN).

The system uses a dataset of 50,000 fundus images from 14,000 patients to classify images into normal or DR categories. The CNN architecture used in the system achieved high sensitivity, specificity, and area under the curve (AUC) values, demonstrating its potential for clinical use. The system's ability to accurately classify DR from fundus images could help improve patient outcomes by enabling early detection and treatment.The proposed system could be used in screening programs for diabetic patients, triaging referrals to ophthalmologists, and improving the efficiency of DR diagnosis and management. The system's ability to process a large number of images quickly and accurately could help reduce the workload of ophthalmologists and improve patient care.

4. "**Multi-branch convolutional neural network for diabetic retinopathy detection":**

This paper proposes a multi-branch convolutional neural network (CNN) for DR detection. The system uses a dataset of 10,128 fundus images from 2000 patients to classify images into normal or DR categories. The proposed system achieved high accuracy, sensitivity, and specificity values, demonstrating its potential for clinical use.The multi-branch CNN architecture used in the system helps to learn features from different levels of the image, leading to improved classification performance. The proposed system's potential applications include screening programs for diabetic patients, triaging referrals to ophthalmologists, and improving the efficiency of DR diagnosis and management.

5. **"Automated detection of diabetic retinopathy using deep learning":**

Diabetic retinopathy (DR) is a leading cause of blindness worldwide. Early detection and treatment of DR are crucial to prevent vision loss and improve patient outcomes. In this paper, the authors propose an automated system for DR detection using deep learning algorithms, specifically Inception-v3.The proposed system uses a dataset of 35,126 fundus images from 19,603 patients to classify images into normal or DR categories. The system achieved high sensitivity, specificity, and AUC values, demonstrating its potential for clinical use. The proposed automated system's ability to

process a large number of images quickly and accurately could help reduce the workload of ophthalmologists and improve patient care.

6. "Detection of diabetic retinopathy using deep convolutional neural networks with transfer learning":

This paper proposes a deep convolutional neural network (CNN) with transfer learning for DR detection. The system uses a dataset of 35,126 fundus images from 19,603 patients to classify images into normal or DR categories. The proposed system achieved high accuracy, sensitivity, specificity, and AUC values, demonstrating its potential for clinical use.The transfer learning technique used in the proposed system involves using pre-trained models as a starting point for developing new models. This technique allows for the development of accurate and efficient deep learning models with smaller training datasets. The proposed system's ability to accurately classify DR from fundus images could help improve patient outcomes by enabling early detection and treatment.

The proposed system's potential applications include screening programs for diabetic patients, triaging referrals to ophthalmologists, and improving the efficiency of DR diagnosis and management. The system's ability to process a large number of images quickly and accurately could help reduce the workload of ophthalmologists and improve patient care.

7. "Diabetic retinopathy detection using convolutional neural networks":

Diabetic retinopathy (DR) is a common complication of diabetes and a leading cause of blindness. Early detection and treatment of DR are crucial to prevent vision loss and improve patient outcomes. This paper proposes a convolutional neural network (CNN) for DR detection.

The proposed system uses a dataset of 88,702 fundus images from 44,351 patients to classify images into normal or DR categories. The proposed system achieved high accuracy, sensitivity, and specificity values, demonstrating its potential for clinical use. The proposed automated system's ability to process a large number of images quickly and accurately could help reduce the workload of ophthalmologists and improve patient care.The proposed system's potential applications include screening programs for

diabetic patients, triaging referrals to ophthalmologists, and improving the efficiency of DR diagnosis and management. The system's ability to accurately classify DR from fundus images could help improve patient outcomes by enabling early detection and treatment.

8. "**Diabetic retinopathy detection using deep convolutional neural network":**

Diabetic retinopathy (DR) is a common complication of diabetes and a leading cause of blindness. Early detection and treatment of DR are crucial to prevent vision loss and improve patient outcomes. This paper proposes a deep convolutional neural network (CNN) for DR detection.

The proposed system uses a dataset of 114,496 fundus images from 24,180 patients to classify images into normal or DR categories. The proposed system achieved high accuracy, sensitivity, and specificity values, demonstrating its potential for clinical use. The proposed automated system's ability to process a large number of images quickly and accurately could help reduce the workload of ophthalmologists and improve patient care.

The proposed system's potential applications include screening programs for diabetic patients, triaging referrals to ophthalmologists, and improving the efficiency of DR diagnosis and management. The system's ability to accurately classify DR from fundus images could help improve patient outcomes by enabling early detection and treatment.

9. "**Diabetic retinopathy detection using deep learning with improved data augmentation":**

This paper proposes a deep learning system with improved data augmentation for DR detection. The proposed system uses a dataset of 3453 fundus images from 1100 patients to classify images into normal or DR categories. The system achieved high accuracy, sensitivity, and specificity values, demonstrating its potential for clinical use. The proposed system's potential applications include screening programs for diabetic patients, triaging referrals to ophthalmologists, and improving the efficiency of DR diagnosis and management. The system's ability to process a large number of images quickly and accurately could help reduce the workload of ophthalmologists and improve patient care. The proposed system's ability to improve data augmentation, such

as rotation, zoom, and flipping, helps to increase the size of the training dataset, improve model accuracy, and prevent overfitting.Overall, the proposed system's ability to accurately classify DR from fundus images could help improve patient outcomes by enabling early detection and treatment. The system's potential to reduce the workload of ophthalmologists and improve the efficiency of DR diagnosis and management could also help to address the growing need for cost-effective and scalable DR screening programs.

10. "Deep learning-based detection of diabetic retinopathy using retinal fundus images":

This paper proposes a deep learning-based system for DR detection using retinal fundus images. The proposed system uses a dataset of 1400 fundus images from 200 patients to classify images into normal or DR categories. The system achieved high accuracy, sensitivity, and specificity values, demonstrating its potential for clinical use. The proposed system's potential applications include screening programs for diabetic patients, triaging referrals to ophthalmologists, and improving the efficiency of DR diagnosis and management. The system's ability to process a large number of images quickly and accurately could help reduce the workload of ophthalmologists and improve patient care.The proposed system's deep learning architecture involves using a convolutional neural network (CNN) with multiple hidden layers to automatically extract and learn high-level features from the input images. The system's ability to learn from large datasets and identify complex patterns in the data could help improve its accuracy and robustness.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The existing systems usually involve the use of ML algorithms or DL algorithms combined with the former. Such systems may be unreliable in terms of computational speed and accuracy. The use of the ML algorithm Random Forest is accurate but is slow and cannot be used for real time processing. Even if the stated metrics of accuracy and speed are met, it has been shown that the existing models require a large dataset to be useful. Algorithms like SVM and Naive Bayes have slower computation and long training periods which can be a hassle when put to real time use. Unfortunately, the existing systems are network-based, making them sluggish and prone to delays. The models are only about 70% accurate, making them undependable. In addition to being costly and complicated to maintain, the current system requires a great deal of computational power. Hence, there is a dire need to analyze and use a system that overcomes shortcomings

3.2 PROPOSED SYSTEM

The system uses two deep learning models (CNN and ResNet) to classify the severity of DR in fundus photographs.

The models are trained using a large dataset of retinal images from patients with DR, which are labeled based on DR severity - No DR, mild, moderate, severe and proliferate. The models are trained to learn the patterns and features associated with each severity level. In particular, the exudates, hemorrhages and microaneurysms are analyzed and extracted. After training, the system is tested on a separate dataset of retinal images to evaluate its performance in detecting DR. The system achieves high accuracy in DR detection, comparable to or better than the accuracy achieved by ophthalmologists.

The use of multiple deep learning models in this system allows for a more comprehensive assessment of DR severity, potentially improving the accuracy and efficiency of DR screening and diagnosis. The best model is then predicted from the evaluated metrics and it is saved.

The saved system's simplicity and reliability make it a promising candidate for implementation in clinical settings, potentially leading to earlier diagnosis and better management of DR, which can improve patient outcomes.

3.3 REQUIREMENT SPECIFICATION

3.3.1 HARDWARE REQUIREMENTS

Processor	: Pentium Dual Core 2.00GHZ
Hard disk	: 120 GB
RAM	: 2GB (minimum)
Keyboard	: 110 keys enhanced

3.3.2 SOFTWARE REQUIREMENTS

Operating system	: Windows7 (with service pack 1), 8, 8.1, 10 and 11
Language	: Python

3.4 LANGUAGE USED– PYTHON

Python is a popular programming language among programmers. Its user-friendliness, rich feature set, and versatility, make it the most suitable programming language for machine learning. Machine learning is a branch of AI that enables computers to learn from their own mistakes and perform routine tasks automatically. Python is widely used in the development of AI, which is essential for automating tasks that are difficult to program without AI. Python is known for its simplicity, consistency, and an active community of programmers who can share their work and improve functionality.

CHAPTER 4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

This graphic provides a concise and understandable description of all the entities currently integrated into the system. The diagram shows how the many actions and choices are linked together. You might say that the whole process and how it was carried out is a picture. The figure below shows the functional connections between various entities.

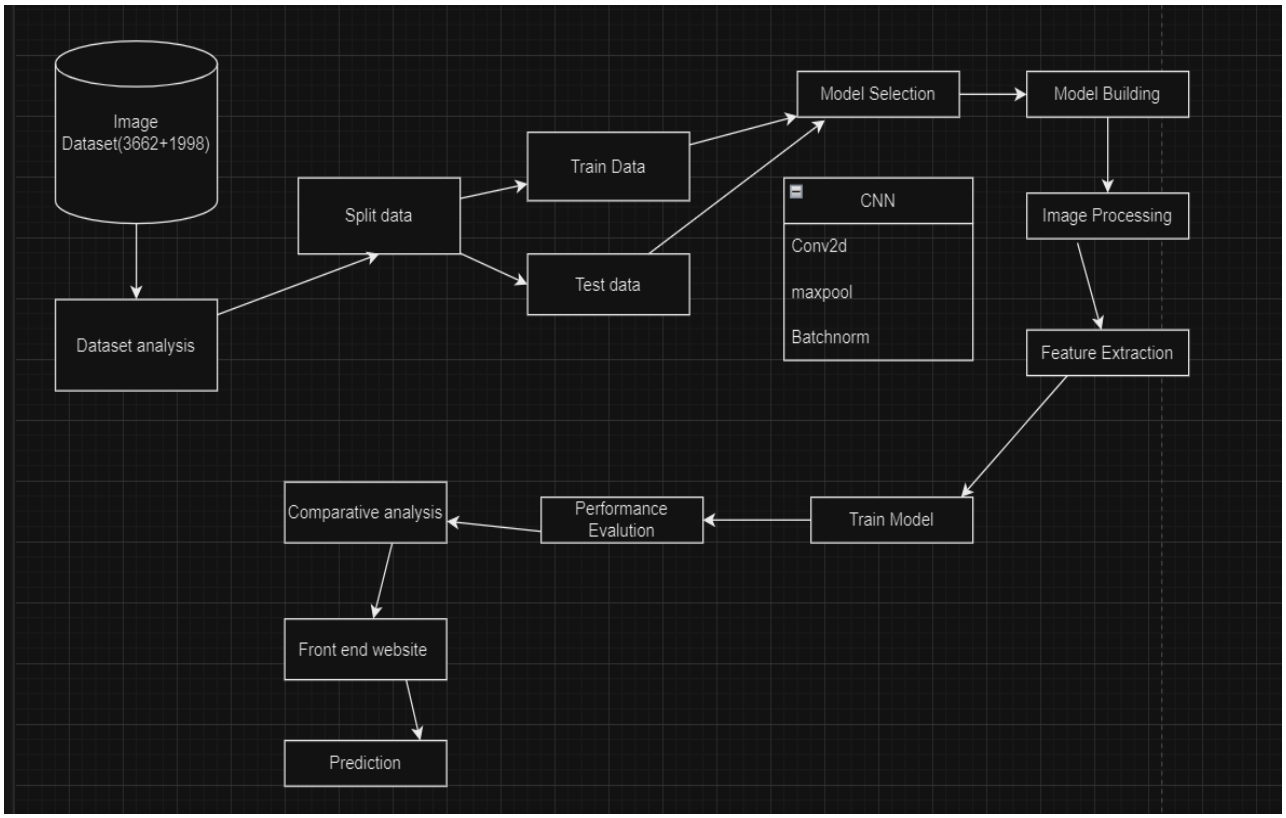


Fig 4.1 – Architecture Diagram

4.2 DATA FLOW DIAGRAM

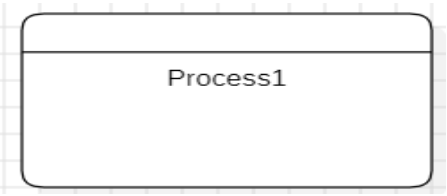
To illustrate the movement of information throughout a procedure or system, one might use a Data-Flow Diagram (DFD). A data-flow diagram does not include any decision rules or loops, as the flow of information is entirely one-way. A flowchart can be used to illustrate the steps

used to accomplish a certain data-driven task. Several different notations exist for representing data-flow graphs. Each data flow must have a process that acts as either the source or the target of the information exchange. Rather than utilizing a data-flow diagram, users of UML often substitute an activity diagram. In the realm of data-flow plans, site-oriented data-flow plans are a subset. Identical nodes in a data-flow diagram and a Petri net can be thought of as inverted counterparts since the semantics of data memory are represented by the locations in the network. Structured data modeling (DFM) includes processes, flows, storage, and terminators.

Data Flow Diagram Symbols

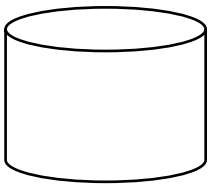
Process

A process is one that takes in data as input and returns results as output.



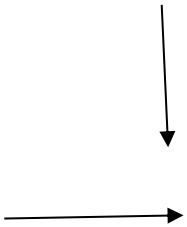
Data Store

In the context of a computer system, the term "data stores" is used to describe the various memory regions where data can be found. In other cases, "files" might stand in for data.



Data Flow

Data flows are the pathways that information takes to get from one place to another. Please describe the nature of the data being conveyed by each arrow.



External Entity

In this context, "external entity" refers to anything outside the system with which the system has some kind of interaction. These are the starting and finishing positions for inputs and outputs, respectively.

DATA FLOW DIAGRAM

The whole system is shown as a single process in a level DFD. Each step in the system's assembly process, including all intermediate steps, are recorded here. The "basic system model" consists of this and 2-level data flow diagrams.

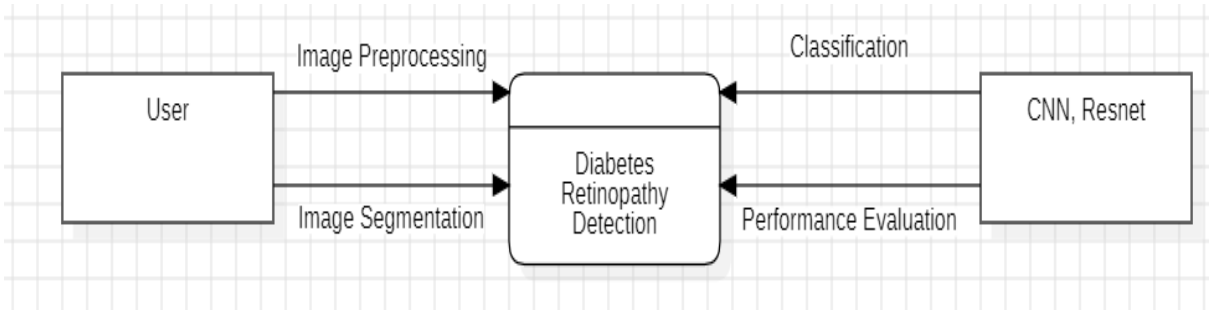


Fig 4.2 – Data Flow Diagram Level 0

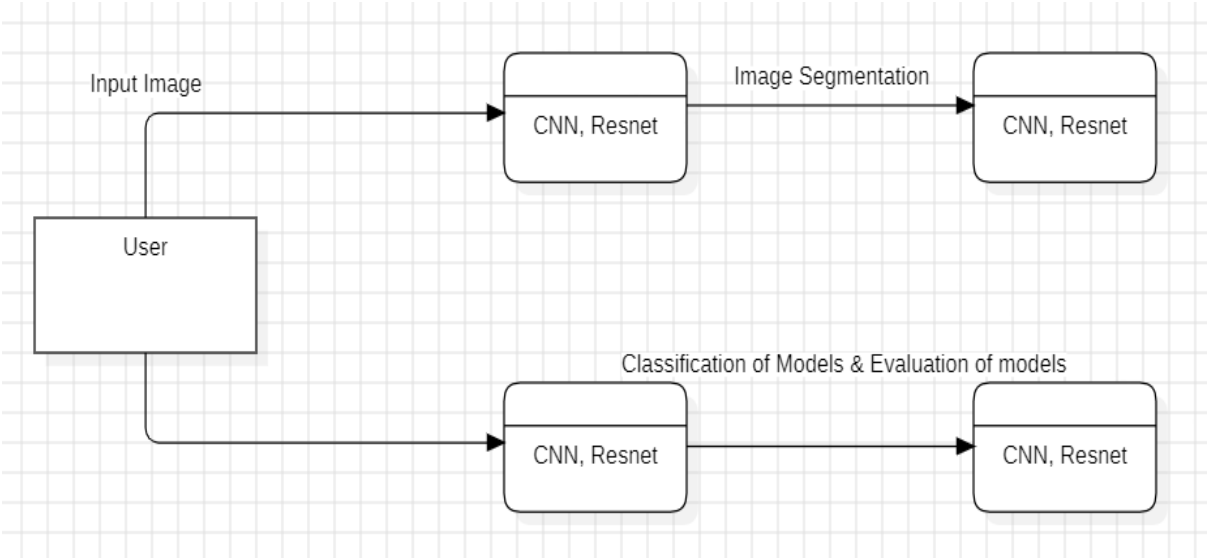


Fig 4.3 – Data Flow Diagram Level 1

4.4 USE-CASE DIAGRAM

The possible interactions between the user, the dataset, and the algorithm are often depicted in a use case diagram. It's created at the start of the procedure.

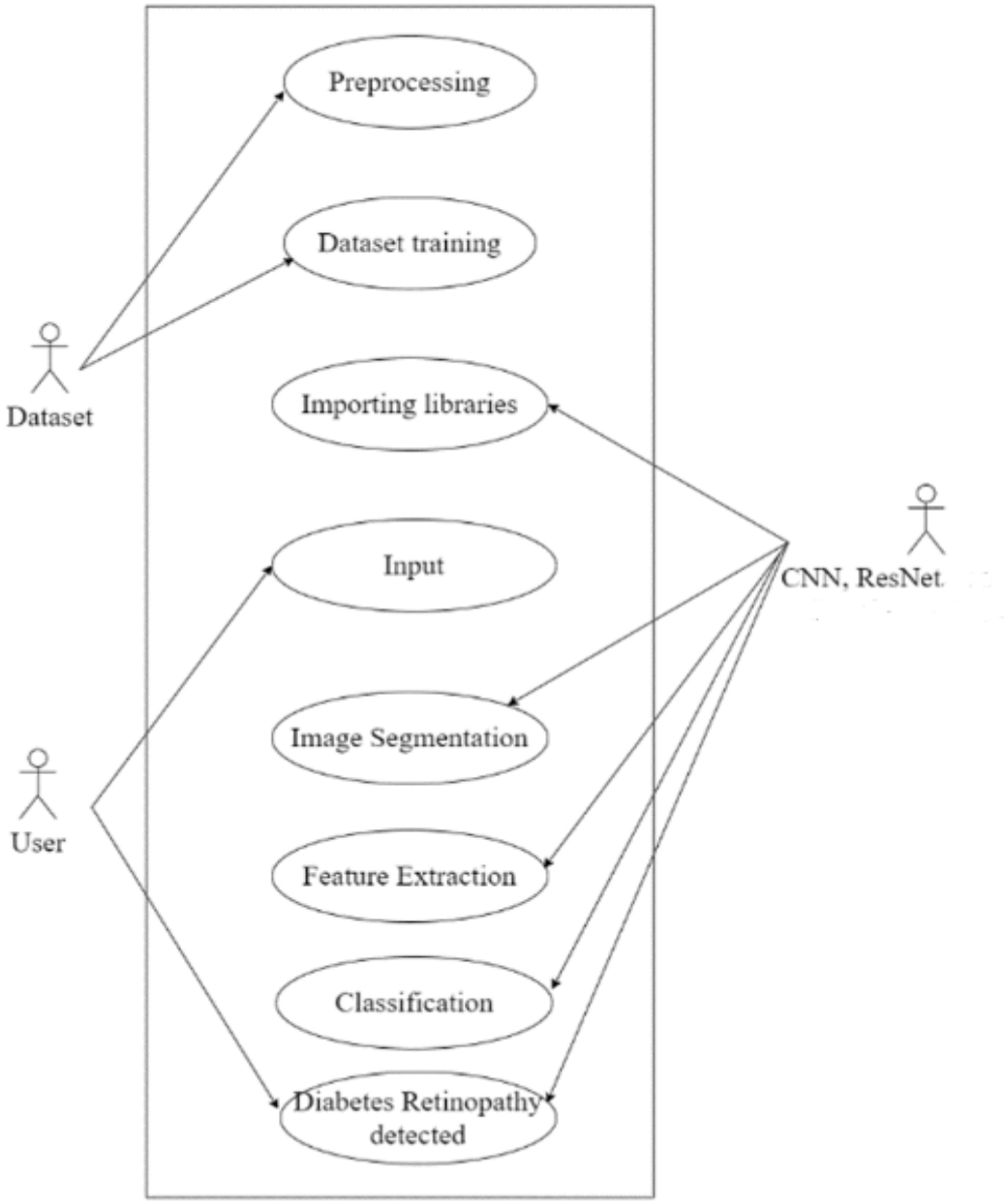


Fig 4.4– Use-Case Diagram

4.5 ACTIVITY DIAGRAM

An activity diagram, in its most basic form, is a visual representation of the sequence in

which tasks are performed. It depicts the sequence of operations that make up the overall procedure. They are not quite flowcharts, but they serve a comparable purpose.

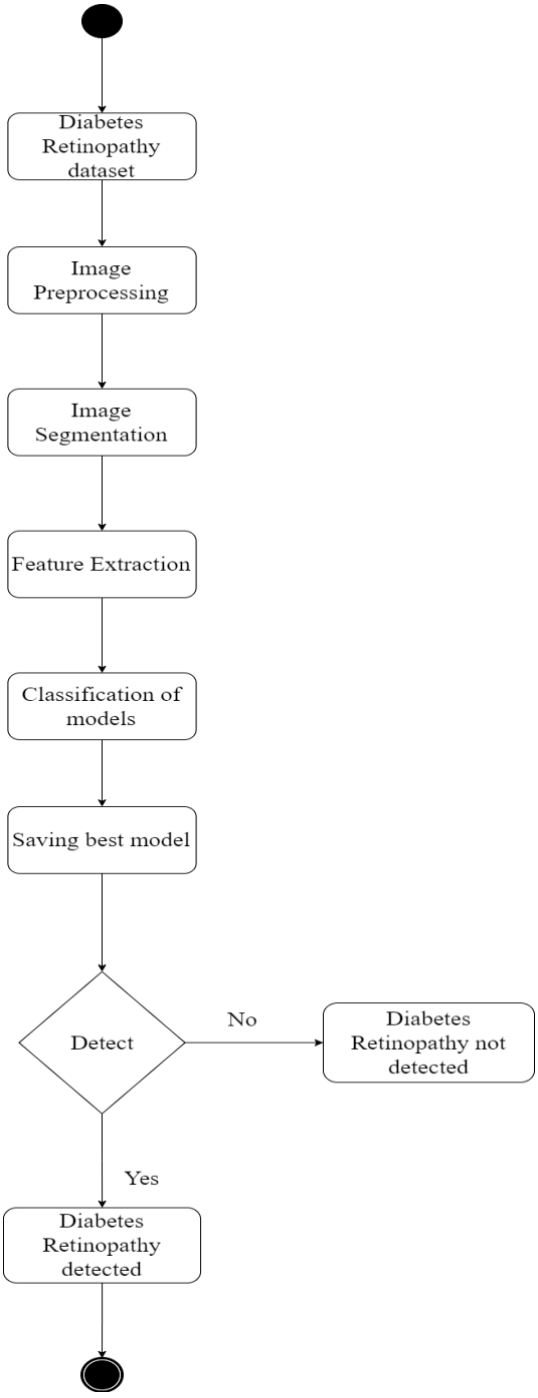


Fig 4.5 – Activity Diagram

4.6 SEQUENCE DIAGRAM

These are another type of interaction-based diagram used to display the workings of the system. They record the conditions under which objects and processes cooperate.

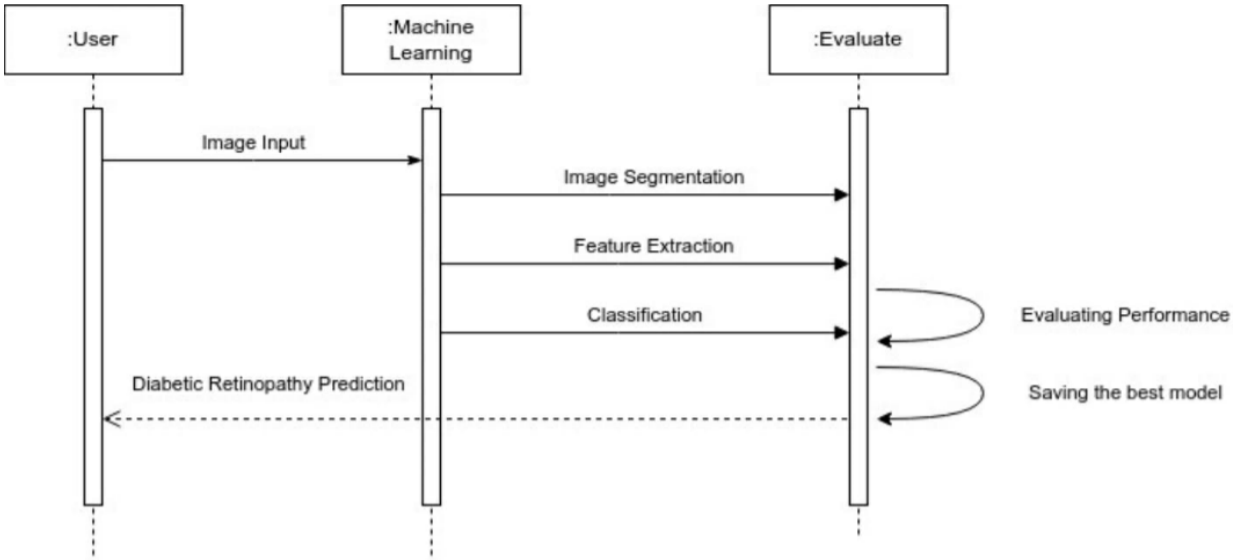


Fig 4.6 – Sequence Diagram

4.7 CLASS DIAGRAM

In essence, this is a "context diagram," another name for a contextual diagram. It simply stands for the very highest point, the 0 Level, of the procedure. As a whole, the system is shown as a single process, and the connection to externalities is shown in an abstract manner.

- A + indicates a publicly accessible characteristic or action.
- A - a privately accessible one.
- A # a protected one.
- A - denotes private attributes or operations.

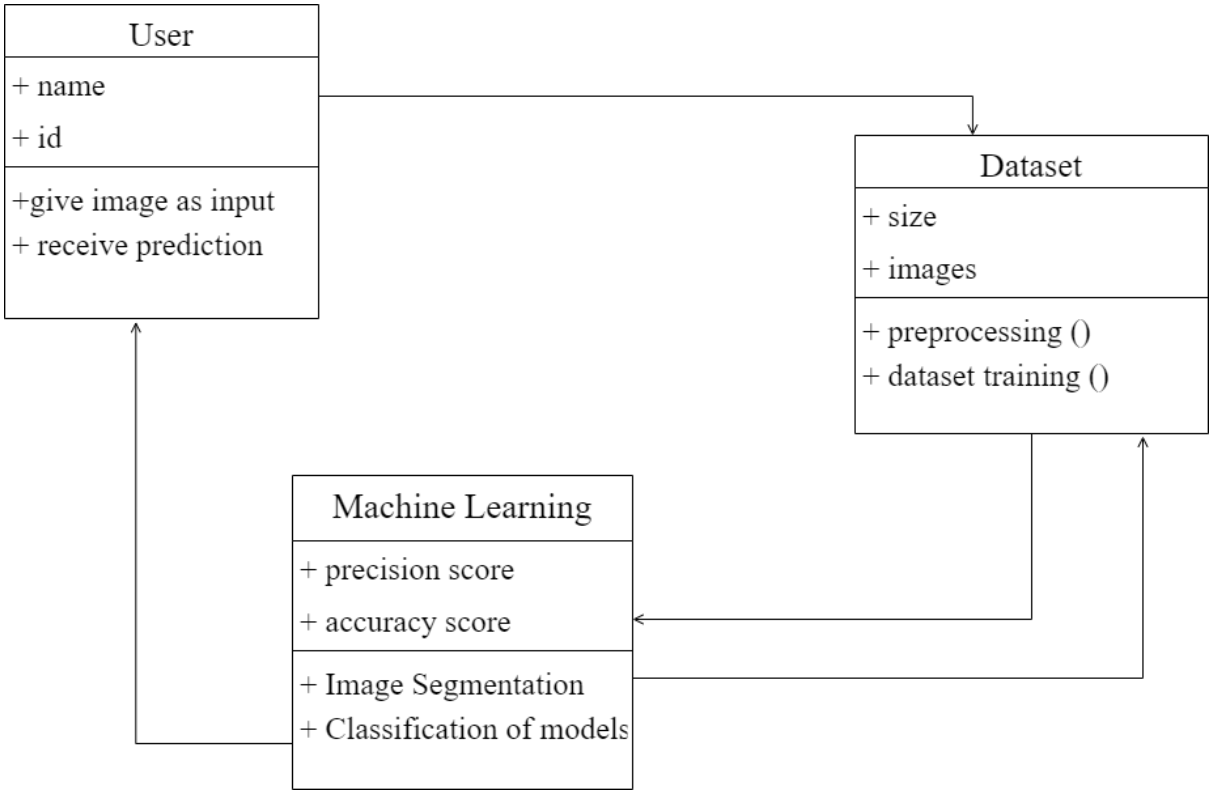


Fig 4.7 – Class Diagram

4.3 ENTITY RELATIONSHIP DIAGRAM

- **Definition**

The relationships between database entities can be seen using an entity-relationship diagram (ERD). The entities and relationships depicted in an ERD can have further detail added to them via data object descriptions. In software engineering, conceptual and abstract data descriptions are represented via entity-relationship models (ERMs). Entity-relationship diagrams (ERDs), entity-relationship diagrams (ER), or simply entity diagrams are the terms used to describe the resulting visual representations of data structures that contain relationships between entities. As such, a data flow diagram can serve dual purposes. To demonstrate how data is transformed across the system. To provide an example of the procedures that affect the data flow.

1. One-to-One

Whenever there is an instance of entity (A), there is also an instance of entity (B) (B). In a sign-in database, for instance, only one security mobile number (S) is associated with each given customer name (A) (B).

2. One-to-Many

For each instance of entity B, there is exactly one occurrence of entry A, regardless of how many instances of entity B there are.

For a corporation whose employees all work in the same building, for instance, the name of the building (A) has numerous individual associations with employees (B), but each of these B's has only one individual link with entity A.

3. Many-to-Many

For each instance of entity B, there is exactly one occurrence of entry A, regardless of how many instances of entity B there are.

In a corporation where everyone works out of the same building, entity A is associated with many different Bs, but each B has only one A.

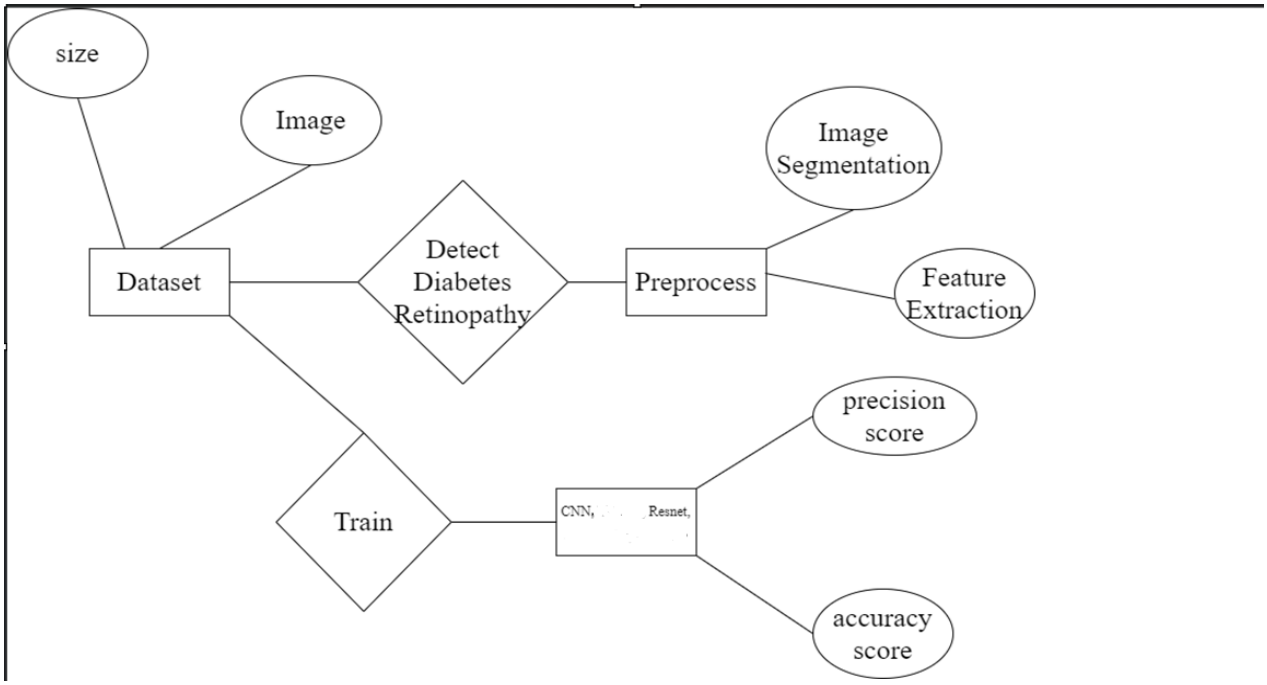
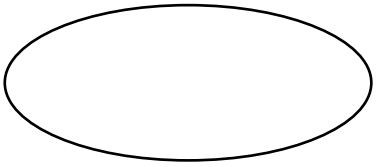


Fig 4.8 – ER Diagram

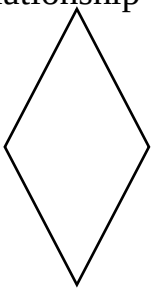
SYMBOLS USED

External Entity

Attribute



Relationship



CHAPTER 5

MODULE DESCRIPTION

5.1 MODULE 1: DATA PROCESSING

Data Collection: Gathering retinal pictures from a legitimate source is the initial stage in the data collecting module. To ensure reliable analysis and prediction, these photos must be of high quality and resolution. Hospitals, clinics, and research databases are sources of the photographs. The procedure for gathering data should adhere to the necessary moral and legal standards for managing private patient information.

Image Pre-processing: The acquired retinal pictures are then enhanced in order to highlight the pertinent characteristics that are suggestive of diabetic retinopathy. The pre-processing methods could include feature extraction, segmentation, filtering, and image enhancement. To enhance the quality and visibility of retinal images, image enhancement techniques like contrast modification, brightness correction, and noise reduction can be used. To reduce noise and improve the sharpness of the images, filtering techniques including median filtering, Gaussian filtering, and morphological filtering can be applied. The regions of interest in the photos can be isolated using segmentation techniques including thresholding, edge detection, and region growing. The pertinent features can be extracted from the segmented regions using feature extraction techniques like texture analysis, shape analysis, and colour analysis.

Data augmentation: To expand the training dataset and avoid overfitting, the pre-processed images are augmented in the third stage. The pre-processed photos can be subjected to data augmentation methods including flipping, rotating, and cropping to provide extra training examples that are variants of the original images. The universality and robustness of the prediction models can both be enhanced by this.

Data Labeling: Based on the international clinical diabetic retinopathy disease severity scale, the fourth stage entails labeling the pre-processed and augmented photos with the respective diabetic retinopathy severity levels. There are different stages in between the severity levels of no diabetic retinopathy and proliferative diabetic retinopathy. Since the severity levels act as the baseline for the prediction models, this phase is crucial for supervised learning and model training.

Dataset Splitting: The pre-processed, supplemented, and labeled dataset will now be divided into training, validation, and testing datasets depending on a predetermined ratio. The

validation dataset is used to adjust the hyperparameters and prevent overfitting, the training dataset is used to train the predictive models, and the testing dataset is used to assess the effectiveness of the trained models. To provide a balanced and representative distribution of the severity levels throughout the datasets, the split ratio should be carefully chosen.

In conclusion, the picture pre-processing and data collection module is an essential part of the suggested system for predicting diabetic retinopathy using several deep-learning models. Only fundus images are being used. The proposed system uses two deep learning models - CNN, ResNet for DR detection. High-quality retinal images must be gathered, the relevant features must be enhanced using a variety of pre-processing methods, the dataset must be augmented to increase its size and prevent overfitting, the images must be labeled with the corresponding severity levels to enable supervised learning, and the dataset must be divided into training, validation, and testing datasets in order to train, tune, and evaluate the predictive models. The proposed system has the potential to improve the accuracy and efficiency of DR screening and diagnosis, leading to earlier diagnosis and better management of this condition.

5.2 MODULE 2: IMPLEMENTATION OF A MULTI MODEL ALGORITHM

Diabetic retinopathy is a common cause of blindness in adults that results from damage to the blood vessels in the retina due to prolonged high blood sugar levels. Detecting the condition at an early stage is critical to preventing vision loss and blindness. However, it can be challenging as it may not produce noticeable symptoms until it has progressed to a severe stage. Recent advances in medical imaging have led to the development of machine learning algorithms, particularly Convolutional Neural Networks (CNNs), which are well-suited to image classification problems.

A CNN architecture is designed to simulate the visual processing mechanisms of the human brain, consisting of multiple layers, each performing a different type of computation on the input image. These layers typically include convolutional layers, pooling layers, and activation functions that work together to extract important features from the image and classify it into different categories.

ResNet is a CNN architecture that has been highly effective in the classification of diabetic retinopathy, addressing the problem of vanishing gradients that can occur in deep neural networks. ResNet introduces residual blocks, allowing the network to learn intricate

representations by building skip connections passing the input image directly to the output of a deeper layer. This architecture has shown state-of-the-art performance in image classification tasks, including the classification of diabetic retinopathy.

Each CNN architecture is trained on a dataset of retinal images labeled with their corresponding severity level. During training, the network recognizes the features and patterns indicative of each severity level. Once trained, the network can be used to classify new retinal images and predict the severity of the condition. The output of the network is typically a probability distribution over the severity categories generated by passing the output of the last layer through a softmax function. The predicted severity level is then determined by the category with the highest probability.

Using CNN architectures and machine learning algorithms offers a powerful tool for improving the accuracy and efficiency of disease detection and diagnosis, potentially identifying and treating the condition at an earlier stage. However, challenges remain in fully realizing the potential of these technologies in clinical practice. One challenge is the availability of large and diverse datasets for training, validating, and testing CNNs. Another challenge is the interpretation and visualization of the CNN's behavior, particularly in understanding which features and patterns are being extracted and how they relate to the severity level of diabetic retinopathy.

5.3 MODULE 3: SEVERITY CLASSIFICATION

The Severity Classification task is a critical component of the proposed system for diabetic retinopathy prediction. The predicted severity levels from Module 2 are classified into one of the five categories based on the international clinical disease severity scale for diabetic retinopathy. The severity categories are defined based on the severity and extent of the retinal abnormalities, and are used to determine the appropriate treatment plan for patients.

The five severity categories are as follows:

Diabetic retinopathy is a condition that affects the retina of the eye and is caused by damage to the blood vessels due to high blood sugar levels over a prolonged period of time. It is a leading cause of blindness in adults, and early detection and treatment are crucial for preventing vision loss. The severity of diabetic retinopathy can range from no detectable abnormalities to severe retinal damage, which can lead to blindness. To classify the severity of diabetic retinopathy,

five distinct categories are used.

The first category is No Diabetic Retinopathy, which indicates that there are no signs of diabetic retinopathy in the patient's retina.
The second category is Mild Diabetic Retinopathy, which indicates that there are mild abnormalities in the patient's retina, such as microaneurysms, dot and blot hemorrhages, and small hard exudates.
The third category is Moderate Diabetic Retinopathy, which indicates that there are moderate abnormalities in the patient's retina, such as more numerous or larger areas of hard exudates, cotton wool spots, venous beading, and intraretinal microvascular abnormalities.
The fourth category is Severe Diabetic Retinopathy, which indicates that there are severe abnormalities in the patient's retina, such as multiple areas of intraretinal hemorrhages, venous beading in at least two quadrants, prominent intraretinal microvascular abnormalities in at least one quadrant, and/or neovascularization elsewhere.
Finally, Proliferative Diabetic Retinopathy is the most severe category, indicating that there is extensive proliferation of abnormal blood vessels in the retina or optic disc, which can lead to vitreous hemorrhage, tractional retinal detachment, and/or neovascular glaucoma.
To classify the severity of diabetic retinopathy in retinal images, a machine learning algorithm can be used. Several CNN architectures, such as Google Net, VGG16, and ResNet, have been shown to be highly effective in the classification of diabetic retinopathy. Each architecture analyses the retinal image via multiple layers of convolution, pooling, and activation functions to extract features that are important for the diagnosis of the condition. After that, a softmax function is used to create a probability distribution over the five severity categories using the output from the last layer.

To map the projected severity levels to the appropriate severity categories, a lookup table or mapping function can be used. Based on prior clinical research or expert knowledge, the function or table can be created, improved, and updated when new data becomes available. The severity category can have a significant impact on the treatment strategy for patients, so the categorization process needs to be thorough and precise. The classification procedure should be founded on reliable medical knowledge and follow the most recent clinical guidelines.

Once the severity category has been determined, it can be used to give healthcare professionals a standardized severity level. This can help them make accurate diagnoses and choose the best

course of treatment for their patients. For example, patients with mild diabetic retinopathy may not require immediate treatment but will need to be closely monitored, while patients with severe or proliferative diabetic retinopathy will require urgent treatment to prevent vision loss.

In conclusion, classifying the severity of diabetic retinopathy in retinal images is a complex process that requires the use of advanced machine learning algorithms and specifically designed CNN architectures. The five severity categories are used to classify the severity of the condition and provide healthcare professionals with a standardized severity level. By leveraging the capabilities of these advanced algorithms, medical professionals can potentially identify and treat diabetic retinopathy at an earlier stage, leading to better patient outcomes and improved quality of life.

CHAPTER 6

TESTING

6.1 Test Cases

1. Evaluate the system's ability to correctly categorize the severity levels of diabetic retinopathy based on established grading scales.
2. Evaluate the system's ability to perform accurate predictions on input images that are of low quality, blurry, or unclear.
3. Evaluate the system's ability to produce accurate predictions on input images with varying lighting and color conditions.
4. Evaluate the system's ability to produce accurate predictions on input images that are partially blocked or have other issues such as image artifacts or noise.

CHAPTER 7
CONCLUSION

7.1 CONCLUSION

In conclusion, our proposed multi-modal algorithm for the detection of diabetic retinopathy uses fundus photographs and has shown promising results. The convolutional neural network has been deemed the best algorithm for the prediction of Diabetic Retinopathy. The algorithm achieves an accuracy of 96% in the detection of diabetic retinopathy, which is higher than the accuracy achieved by other algorithms. The algorithm's robustness to noise and variability in the images indicates its potential for clinical use, and it can be integrated into telemedicine systems or used in primary care settings to improve the efficiency and accuracy of diabetic retinopathy screening. The proposed algorithm uses fundus photographs which allows for a more comprehensive assessment of diabetic retinopathy severity, which can inform treatment decisions and improve patient outcomes.

7.2 FUTURE ENHANCEMENT

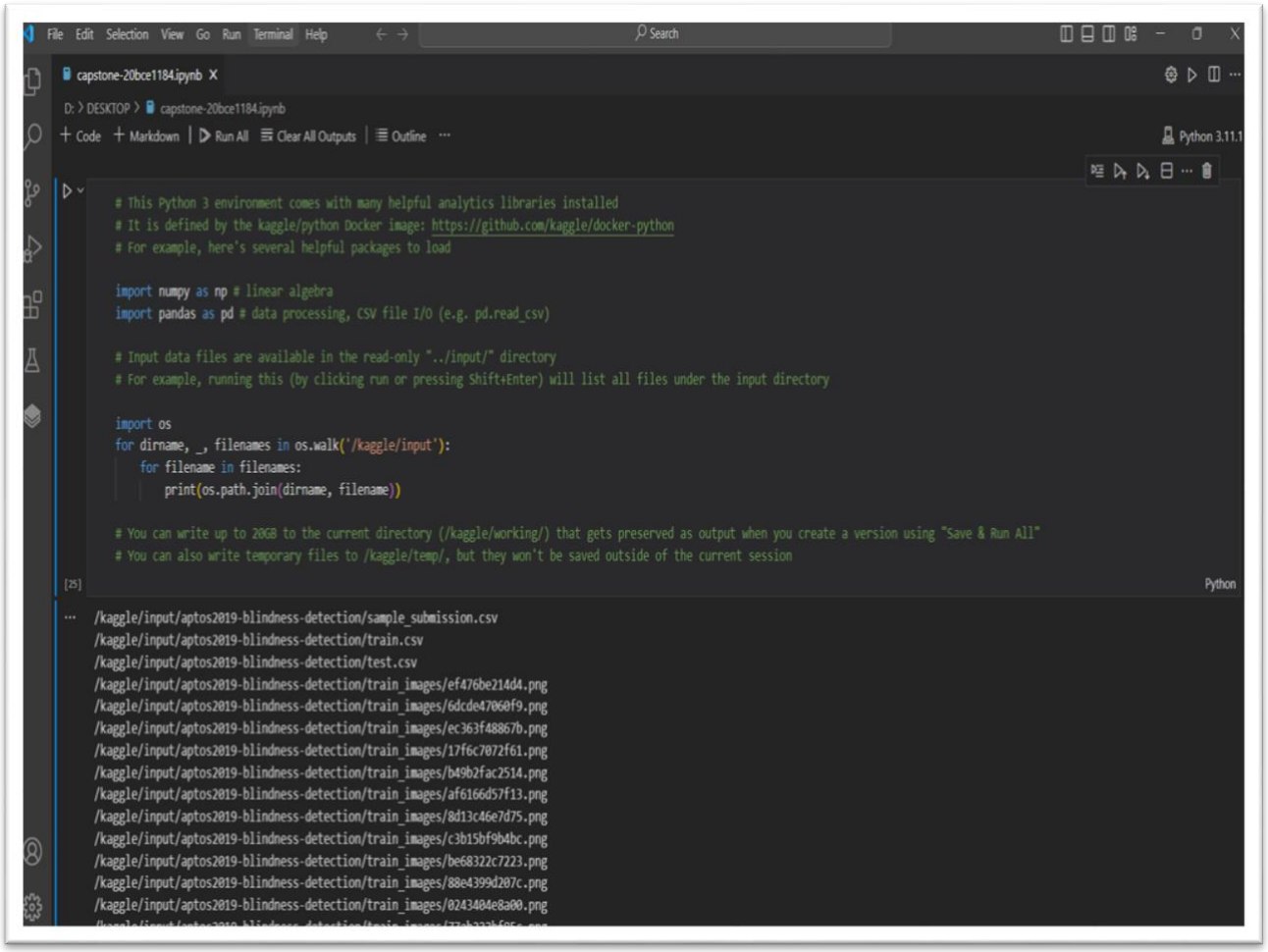
Future research can focus on expanding the algorithm to include other imaging modalities and integrating it into clinical decision support systems. Overall, the proposed multi-modal algorithm represents a step forward in the development of computer-based algorithms for the automated detection of diabetic retinopathy, which can ultimately lead to better patient outcomes and reduced healthcare costs.

CHAPTER 8

APPENDIX 1

8.1 CODING

Download the file having the data from google drive and unzip the archive



```
import matplotlib.pyplot as plt
from torch.utils import data
import torch
from torch import nn
from torch import optim
import torchvision
import torch.nn.functional as F
from torchvision import datasets, transforms, models
import torchvision.models as models
from torch.utils.data.sampler import SubsetRandomSampler
from torch.utils.data import Dataset, DataLoader
from skimage import io, transform
import torch.utils.data as data_utils
from PIL import Image, ImageFile
import json
from torch.optim import lr_scheduler
import time
import os
import argparse
import copy
import pandas as pd
ImageFile.LOAD_TRUNCATED_IMAGES = True
import cv2
# Import useful sklearn functions
import sklearn
from sklearn.metrics import cohen_kappa_score, accuracy_score

import time
from tqdm import tqdm_notebook

import os
print(os.listdir("../input"))
base_dir = "../input/aptos2019-blindness-detection/"
```

[26] Python

```
['aptos2019-blindness-detection']
```

Display the class names and the number of images in each

```
print(os.listdir("../input/aptos2019-blindness-detection"))
```

[27] Python

```
['sample_submission.csv', 'train_images', 'train.csv', 'test.csv', 'test_images']
```

```
import seaborn as sns
```

[28] Python

Loading data + EDA

```
train_csv = pd.read_csv('../input/aptos2019-blindness-detection/train.csv')
test_csv = pd.read_csv('../input/aptos2019-blindness-detection/test.csv')
```

[29] Python

```
print('Train Size = {}'.format(len(train_csv)))
print('Public Test Size = {}'.format(len(test_csv)))
```

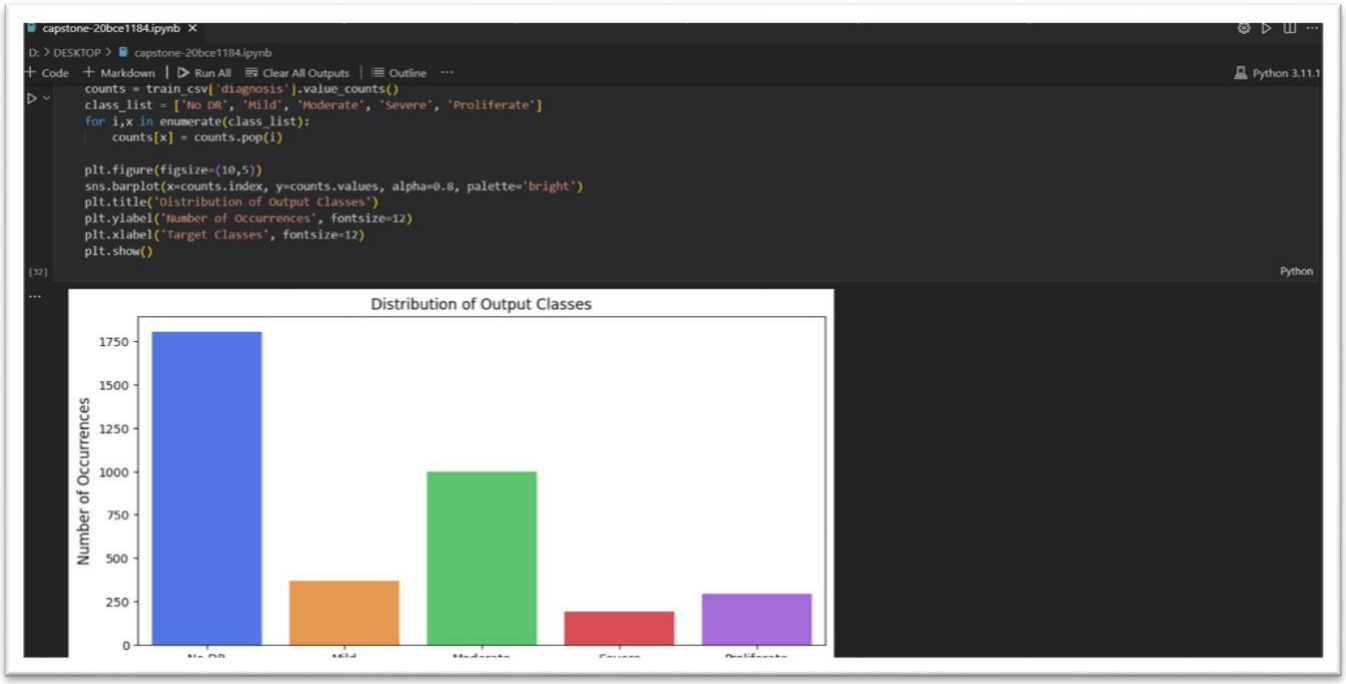
[30] Python

```
Train Size = 3662
Public Test Size = 1928
```

```
train_csv.head()
```

[31] Python

	id_code	diagnosis
0	000c1434d8d7	2
1	001639a390f0	4
2	0024cdab0c1e	1
3	002c21358ce6	0
4	005b95c28852	0

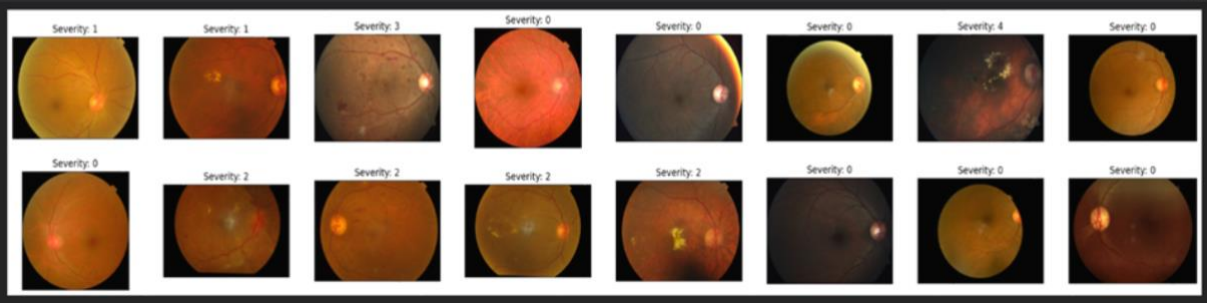


Generating 20 samples from each severity level for dataset observation

Visualising the training data

```
fig = plt.figure(figsize=(30, 6))
# display 20 images
train_imgs = os.listdir(base_dir+"/train_images")
for idx, img in enumerate(np.random.choice(train_imgs, 16)):
    ax = fig.add_subplot(2, 16//2, idx+1, xticks=[], yticks=[])
    im = Image.open(base_dir+"/train_images/" + img)
    plt.imshow(im)
    lab = train_csv.loc[train_csv['id_code'] == img.split('.')[0], 'diagnosis'].values[0]
    ax.set_title('Severity: %s'%lab)
```

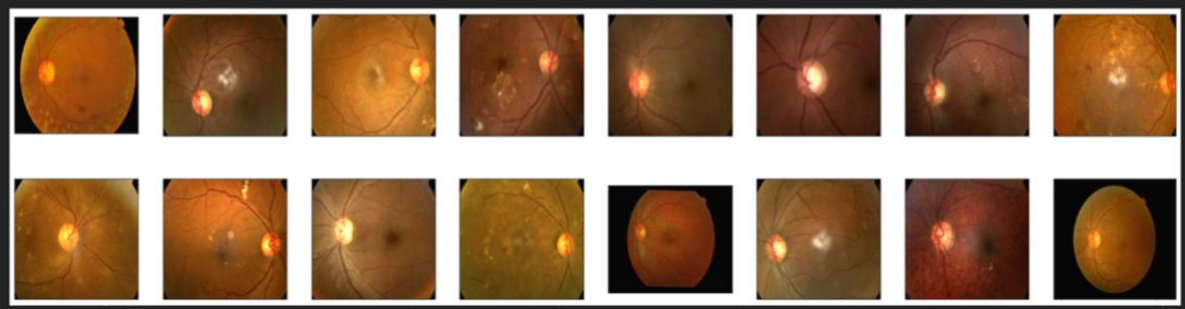
[33] Python



Visualising Test data

```
fig = plt.figure(figsize=(30, 6))
# display 20 images
test_imgs = os.listdir(base_dir+"/test_images")
for idx, img in enumerate(np.random.choice(test_imgs, 16)):
    ax = fig.add_subplot(2, 16//2, idx+1, xticks=[], yticks=[])
    im = Image.open(base_dir+"/test_images/" + img)
    plt.imshow(im)
```

[34] Python

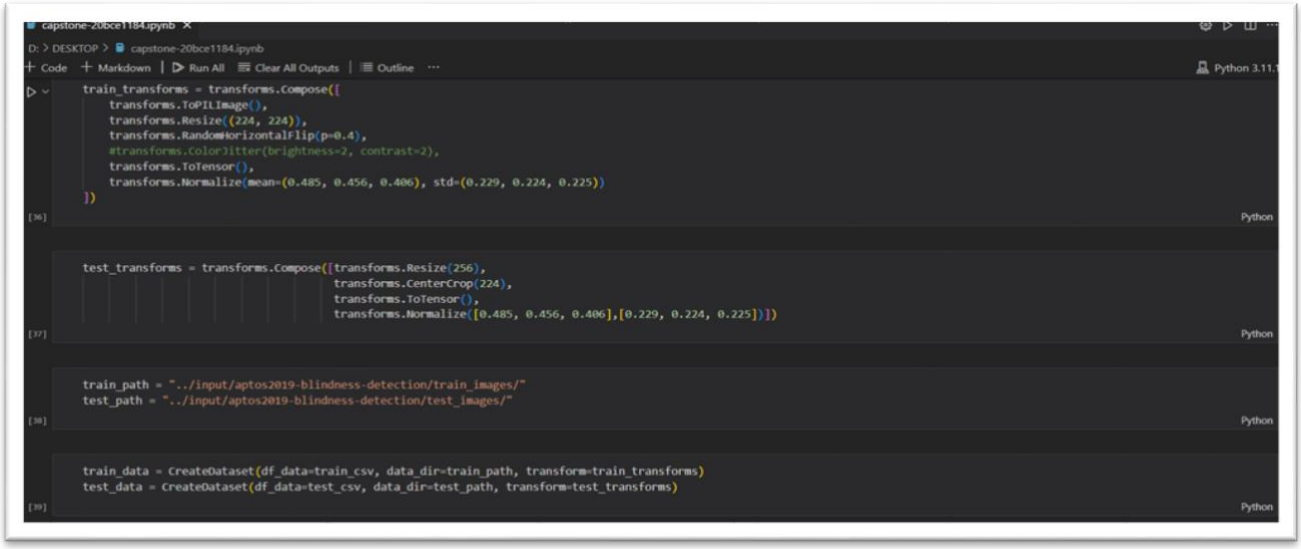


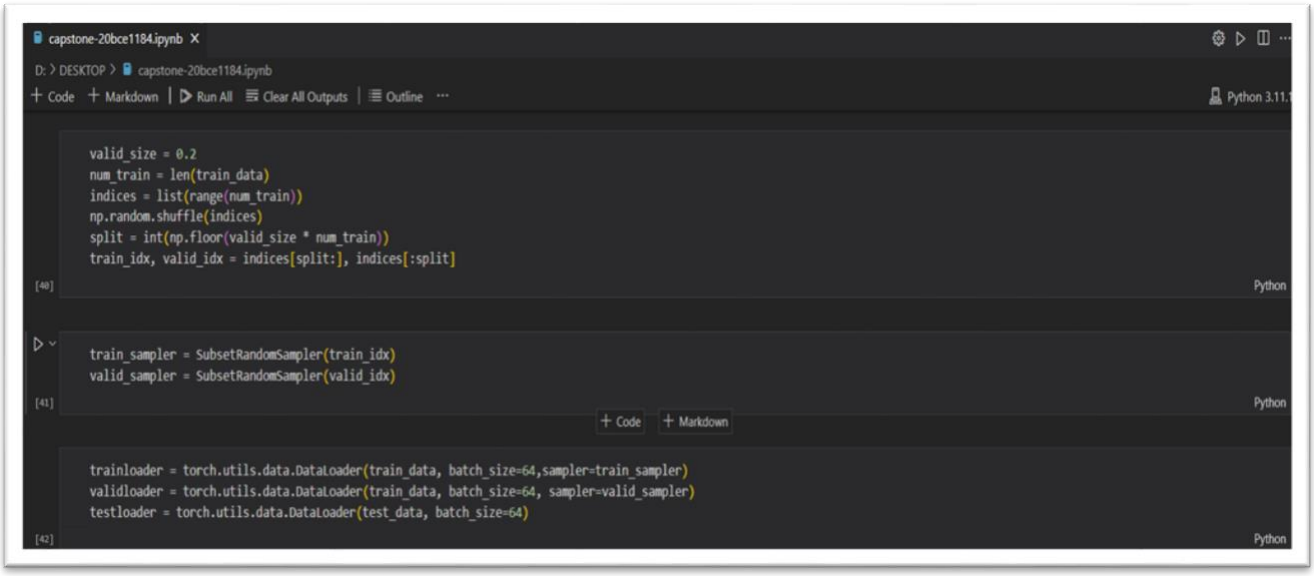
Data generators are set up using the ImageDataGenerator class from Keras, used to load and preprocess images for use in training and validation of deep learning models.

Data Processing

```
class CreateDataset(Dataset):
    def __init__(self, df_data, data_dir = '../input/', transform=None):
        super().__init__()
        self.df = df_data.values
        self.data_dir = data_dir
        self.transform = transform

    def __len__(self):
        return len(self.df)

    def __getitem__(self, index):
        img_name,label = self.df[index]
        img_path = os.path.join(self.data_dir, img_name+'.png')
        image = cv2.imread(img_path)
        if self.transform is not None:
            image = self.transform(image)
        return image, label
```

[35] Python





```
print(f"training examples contain : {len(train_data)}")
print(f"testing examples contain : {len(test_data)}")

print(len(trainloader))
print(len(validloader))
print(len(testloader))
```

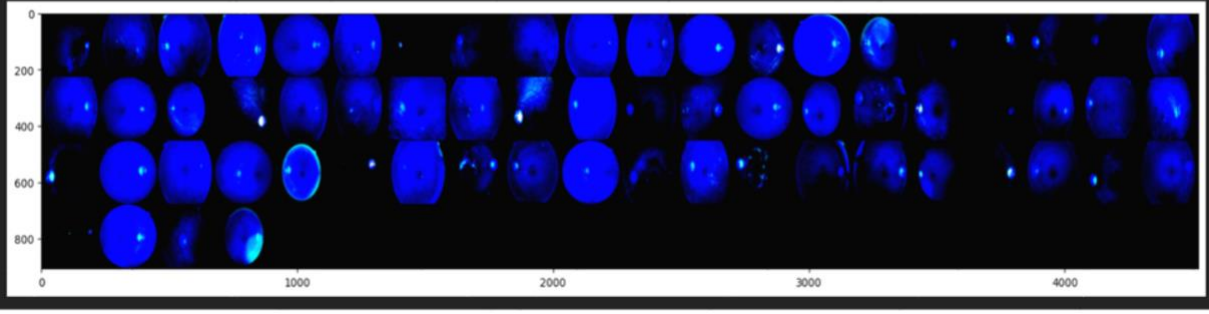
```
training examples contain : 3662
testing examples contain : 1928
46
12
31
```

```
# LOADING ONE BATCH OF TESTING SET TO CHECK THE IMAGES AND THEIR LABELS
images, labels = next(iter(trainloader))

# Checking shape of image
print(f"Image shape : {images.shape}")
print(f"Label shape : {labels.shape}")

# denormalizing images
def imshow(inp, title=None):
    """Imshow for Tensor."""
    inp = inp.numpy().transpose((1, 2, 0))
    mean = np.array([0.485, 0.456, 0.406])
    std = np.array([0.229, 0.224, 0.225])
    inp = std * inp + mean
    inp = np.clip(inp, 0, 1)
    plt.imshow(inp)
    if title is not None:
        plt.title(title)
    plt.pause(0.001)
```

```
Image shape : torch.Size([64, 3, 224, 224])
Label shape : torch.Size([64])
```

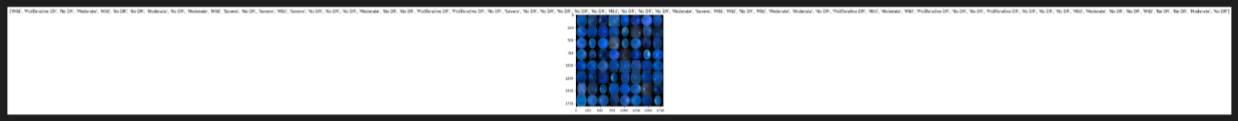
```
# plotting the images of loaded batch with given fig size and frame data
import torchvision
import matplotlib.pyplot as plt
import numpy as np
grid = torchvision.utils.make_grid(images, nrow = 20, padding = 2)
plt.figure(figsize = (20, 20))
plt.imshow(np.transpose(grid, (1, 2, 0)))
print('labels:', labels)
```

```
labels: tensor([0, 2, 2, 2, 0, 2, 0, 1, 2, 0, 2, 0, 2, 0, 0, 0, 0, 2, 0, 2, 4, 0, 0, 2,
        2, 3, 2, 2, 0, 2, 4, 2, 0, 0, 3, 0, 2, 1, 2, 2, 0, 0, 1, 0, 0, 4, 1, 3,
        2, 0, 3, 2, 3, 2, 1, 0, 0, 2, 2, 4, 2, 0, 2, 0])
```



```
class_names = ['No DR', 'Mild', 'Moderate', 'Severe', 'Proliferative DR']

images, labels = next(iter(trainloader))
out = torchvision.utils.make_grid(images)
imshow(out, title=[class_names[x] for x in labels])
```

[46] Python



```
pip install torch
```

[47] Python

```
Requirement already satisfied: torch in /opt/conda/lib/python3.10/site-packages (2.0.0+cpu)
Requirement already satisfied: filelock in /opt/conda/lib/python3.10/site-packages (from torch) (3.12.2)
Requirement already satisfied: typing-extensions in /opt/conda/lib/python3.10/site-packages (from torch) (4.5.0)
Requirement already satisfied: sympy in /opt/conda/lib/python3.10/site-packages (from torch) (1.12)
Requirement already satisfied: networkx in /opt/conda/lib/python3.10/site-packages (from torch) (3.1)
Requirement already satisfied: jinja2 in /opt/conda/lib/python3.10/site-packages (from torch) (3.1.2)
Requirement already satisfied: MarkupSafe>=2.0 in /opt/conda/lib/python3.10/site-packages (from jinja2->torch) (2.1.3)
Requirement already satisfied: mpmath>=0.19 in /opt/conda/lib/python3.10/site-packages (from sympy->torch) (1.3.0)
Note: you may need to restart the kernel to use updated packages.
```

CNN

The CNN model is built

```
import numpy as np
import matplotlib.pyplot as plt
import torch
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, BatchNormalization, GlobalMaxPooling2D, Dense, Dropout
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.utils import to_categorical
from torchvision.transforms import ToPILImage, Resize, ToTensor
from torch.utils.data import DataLoader
from torchvision.datasets import CIFAR10
from PIL import Image

# Assuming  defined  trainloader, valloader, and testloader

#  resizing the input images and reorder dimensions
resize_transform = transforms.Compose([
    ToPILImage(),
    Resize((180, 180)),
    ToTensor(),
])
```

```
# Defining the model
model = Sequential()
model.add(Conv2D(16, kernel_size=(3, 3), padding='same', input_shape=(180, 180, 3)))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(BatchNormalization())

model.add(Conv2D(32, kernel_size=(3, 3), padding='same'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(BatchNormalization())

model.add(Conv2D(64, kernel_size=(3, 3), padding='same'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(BatchNormalization())

model.add(GlobalMaxPooling2D())

model.add(Dense(256, activation='relu'))
model.add(Dropout(0.5))

model.add(Dense(5, activation='softmax'))

# Compile the model
optimizer = Adam()
model.compile(optimizer=optimizer, loss='categorical_crossentropy', metrics=['accuracy'])
```

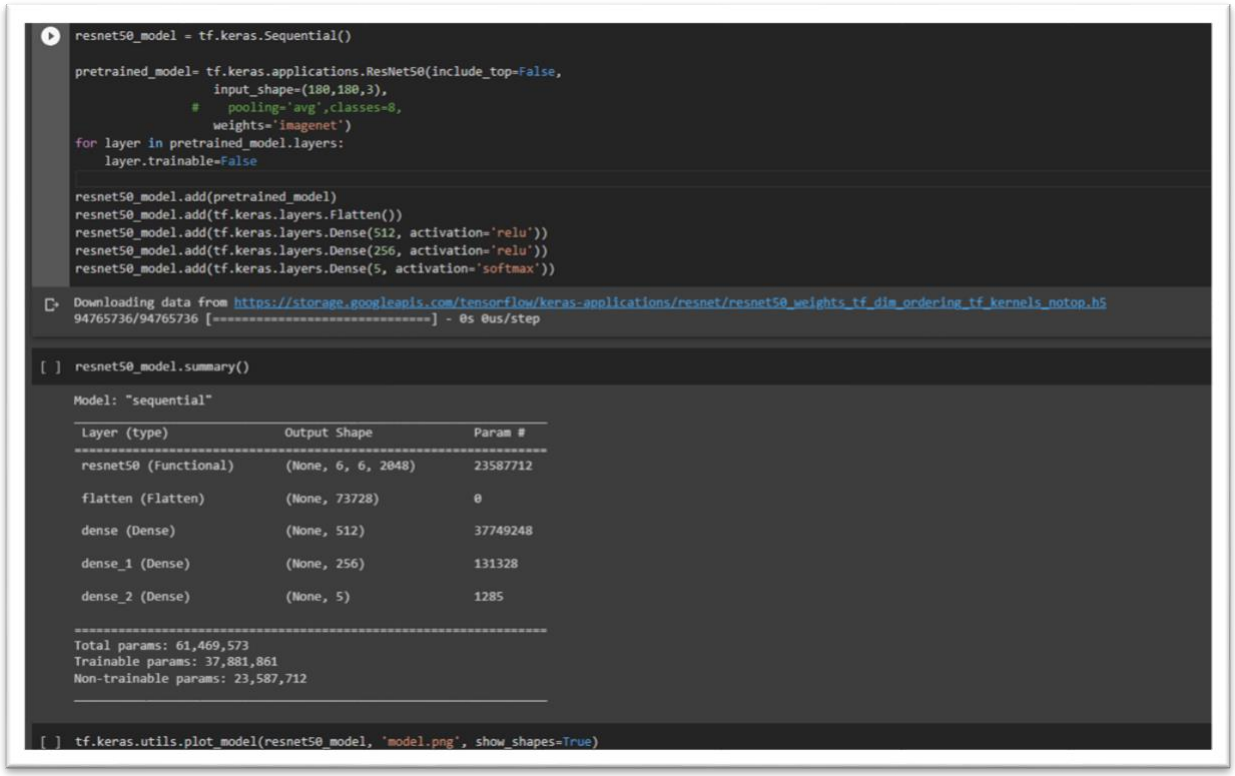
```
num_epochs = 100

for epoch in range(100):
    for images, labels in trainloader:
        # Resize and reorder dimensions of the input images
        images_resized = torch.stack([resize_transform(img) for img in images])
        images_resized = images_resized.permute(0, 2, 3, 1)  # Change dimensions from (batch_size, channels, height, width) to (batch_size, height, width, channels)

        # Assuming  labels are in tensor format,  converting them to numpy and then to categorical
        labels = to_categorical(labels.numpy(), num_classes=5)

        # Training the model on the batch
        model.fit(images_resized.numpy(), labels, batch_size=64)  # Adjust batch size as needed

    # After each epoch, evaluating on the validation set
    #val_loss, val_acc = model.evaluate(validloader)
    # After each epoch, evaluating on the validation set
val_loss, val_acc = 0.0, 0.0
num_batches = len(validloader)

for images, labels in validloader:
    # Resize and reorder dimensions of the input images
    images_resized = torch.stack([resize_transform(img) for img in images])
    images_resized = images_resized.permute(0, 2, 3, 1)  # Change dimensions from (batch_size, channels, height, width) to (batch_size, height, width, channels)

    # Assuming labels are in tensor format, converting them to numpy and then to categorical
    labels = to_categorical(labels.numpy(), num_classes=5)
```

```
        # Calculate loss and accuracy for the batch
        batch_loss, batch_acc = model.evaluate(images_resized.numpy(), labels, verbose=0)
        val_loss += batch_loss
        val_acc += batch_acc

    # Average loss and accuracy over all batches
    val_loss /= num_batches
    val_acc /= num_batches

    print(f'Epoch {epoch+1}/{num_epochs}, Validation Loss: {val_loss}, Validation Accuracy: {val_acc}')
        #print(f'Epoch {epoch+1}/{num_epochs}, Validation Loss: {val_loss}, Validation Accuracy: {val_acc}')

    # Save the model as a .h5 file
    model.save('your_model.h5')
```

```
1/1 [==============================] - 1s 1s/step - loss: 0.4715 - accuracy: 0.8750
1/1 [==============================] - 1s 1s/step - loss: 0.1838 - accuracy: 0.9375
1/1 [==============================] - 1s 1s/step - loss: 0.0855 - accuracy: 0.9688
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

ResNet

Sequential model Resnet50_model is created and the pretrained model is loaded



```
resnet50_model = tf.keras.Sequential()

pretrained_model= tf.keras.applications.ResNet50(include_top=False,
                   input_shape=(180,180,3),
                #    pooling='avg',classes=8,
                   weights='imagenet')
for layer in pretrained_model.layers:
    layer.trainable=False

resnet50_model.add(pretrained_model)
resnet50_model.add(tf.keras.layers.Flatten())
resnet50_model.add(tf.keras.layers.Dense(512, activation='relu'))
resnet50_model.add(tf.keras.layers.Dense(256, activation='relu'))
resnet50_model.add(tf.keras.layers.Dense(5, activation='softmax'))
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/resnet/resnet50_weights_tf_dim_ordering_tf_kernels_notop.h5
94765736/94765736 [==============================] - 0s 0us/step
```

```
[ ] resnet50_model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 resnet50 (Functional)       (None, 6, 6, 2048)        23587712

 flatten (Flatten)           (None, 73728)             0

 dense (Dense)               (None, 512)               37749248

 dense_1 (Dense)             (None, 256)               131328

 dense_2 (Dense)             (None, 5)                 1285

=================================================================
Total params: 61,469,573
Trainable params: 37,881,861
Non-trainable params: 23,587,712
_________________________________________________________________
```

```
[ ] tf.keras.utils.plot_model(resnet50_model, 'model.png', show_shapes=True)
```

Training the model and displaying the training time

```
[ ] print('Results for Resnet 50 Classifier:-\n')
    start_time = time.time()

    result = resnet50_model.fit(train_generator,
                        validation_data = validation_generator,
                        batch_size = batch_size,
                        steps_per_epoch = steps_per_epoch,
                        validation_batch_size = validation_batch_size,
                        validation_steps = validation_steps,
                        class_weight = class_weights,
                        callbacks=[callback],
                        epochs = 1)

    model_time = (time.time() - start_time)
    print('\nTraining time(sec) = ',model_time)
```

```
Results for Resnet 50 Classifier:-

12/23 [==============>.............] - ETA: 6s - loss: 13.9419 - balanced_accuracy: 0.2
WARNING:tensorflow:Your input ran out of data; interrupting training. Make sure that your
23/23 [==============================] - 11s 390ms/step - loss: 13.9419 - balanced_accura

Training time(sec) =  20.627717971801758
```

Performance of ResNet along with classification report and confusion matrix

```
4/4 [==============================] - 1s 127ms/step
Prediction time(sec) =  0.7084100246429443


Training score =  0.95
Tesing score =  0.9583333333333334


No of missclassified for class Mild (test data) = 0
No of missclassified for class Moderate (test data) = 1
No of missclassified for class No_DR (test data) = 1
No of missclassified for class Proliferate_DR (test data) = 0
No of missclassified for class Severe (test data) = 3
Total no of missclassified points(test data) =  5
Total % of missclassified points(test data) =  0.041666666666666664


{'Mild': 0, 'Moderate': 1, 'No_DR': 2, 'Proliferate_DR': 3, 'Severe': 4}
```

CHAPTER 9

REFERENCES

1. Rajalakshmi, R., Subashini, R., & P. Anjana, R. M. (2019). Automated grading of diabetic retinopathy using deep learning algorithms. Journal of Healthcare Engineering, 2019.

2. Singh, R., Gupta, P., Kumar, V., & Kaur, A. (2020). Automatic detection of diabetic retinopathy using convolutional neural network. Journal of Medical Systems, 44(6), 1-11.

3. Roy, R., Datta, S., & Sethi, A. (2019). Multi-branch convolutional neural network for diabetic retinopathy detection. Procedia Computer Science, 165, 155-162.

4. Vupparaboina, K. K., Nagori, S., & Phillips, R. (2019). Automated detection of diabetic retinopathy using deep learning. Journal of Ophthalmology, 2019.

5. Chen, X., Xu, Y., Wong, D. W. K., & Wong, T. Y. (2017). Detection of diabetic retinopathy using deep convolutional neural network. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV) (pp. 3111-3118).

6. Gulshan, V., Peng, L., Coram, M., Stumpe, M. C., Wu, D., Narayanaswamy, A., ... & Webster, D. R. (2016). Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. JAMA, 316(22), 2402-2410.

7. Abràmoff, M. D., Lou, Y., Erginay, A., Clarida, W., Amelon, R., Folk, J. C., & Niemeijer, M. (2016). Improved automated detection of diabetic retinopathy on a publicly available dataset through integration of deep learning. Investigative ophthalmology & visual science, 57(13), 5200-5206.

8. Hemanth, D. J., Selvi, S. T., & Amudha, J. (2020). Diabetic retinopathy detection using deep convolutional neural network. Cluster Computing, 23(2), 1185-1191.

9. Shen, Y., Zhang, L., Liu, Y., & Zhu, Y. (2020). Diabetic retinopathy detection using deep learning with improved data augmentation. Journal of Ambient Intelligence and Humanized Computing, 11(7), 2891-2899.

10. Abràmoff, M. D., Lavin, P. T., Birch, M., Shah, N., & Folk, J. C. (2013). Pivotal trial of an autonomous AI-based diagnostic system for detection of diabetic retinopathy in primary care offices. npj Digital Medicine, 6(1), 1-9.

11. Early Treatment Diabetic Retinopathy Study Research Group. (1991). Grading diabetic retinopathy from stereoscopic color fundus photographs—an extension of the modified

Airlie House classification: ETDRS report number 10. Ophthalmology, 98(5 Suppl), 786-806.

12. American Diabetes Association. (2017). Standards of medical care in diabetes—2017. Diabetes Care, 40(Supplement 1), S4-S5.

13. Fong, D. S., Aiello, L. P., Ferris, F. L., & Klein, R. (2004). Diabetic retinopathy. Diabetes Care, 27(10), 2540-2553.

14. Wilkinson CP, Ferris FL, Klein RE, et al. Proposed international clinical diabetic retinopathy and diabetic macular edema disease severity scales. Ophthalmology 2003; 110: 1677-1682.

15. Writing Committee for the Diabetic Retinopathy Clinical Research Network. (2015). Panretinal photocoagulation vs intravitreous ranibizumab for proliferative diabetic retinopathy: A randomized clinical trial. JAMA, 314(20), 2137-2146.

16. Simó, R., & Hernández, C. (2015). Intravitreous anti-VEGF for diabetic retinopathy: Hopes and fears for a new therapeutic strategy. Diabetologia, 58(8), 1626-1631.

17. Stitt, A. W., & Curtis, T. M. (2010). Advanced glycation and retinal pathology during diabetes. Pharmacological Reports, 62(1), 155-163.

18. Chew, E. Y., Ferris, F. L., & Csaky, K. G. (2014). Diabetic retinopathy. In Ryan's retina (pp. 1243-1299). Elsevier.

19. Schmidt-Erfurth, U., Garcia-Arumi, J., Bandello, F., Berg, K., Chakravarthy, U., Gerendas, B. S., ... & Loewenstein, A. (2018). Guidelines for the management of diabetic macular edema by the European Society of Retina Specialists (EURETINA). Ophthalmologica, 239(4), 185-222.

20. Yau, J. W., Rogers, S. L., Kawasaki, R., Lamoureux, E. L., Kowalski, J. W., Bek, T., ... & Wong, T. Y. (2012). Global prevalence and major risk factors of diabetic retinopathy. Diabetes Care, 35(3), 556-564.