A RESNET-50 MODEL FOR THE DETECTION OF DIABETIC RETINOPATHY

A

Project Report Submitted

In partial fulfillment of the requirements for the award of the Degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE & ENGINEERING

By

Patiballa Surya Vamsi **19761A0547**

Under the esteemed guidance of

Mr. N V Naik

Sr. Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE &ENGINEERING

LAKIREDDY BALIREDDY COLLEGE OF ENGINEERING

(AUTONOMOUS)

Accredited by NAAC with 'A' Grade & NBA (Under Tier - I), ISO 9001:2015 Certified Institution Approved by AICTE, New Delhi and Affiliated to JNTUK, Kakinada

L.B. REDDY NAGAR, MYLAVARAM, NTR DIST., A.P.-521 230.

2019-2023

LAKIREDDY BALI REDDY COLLEGE OF ENGINEERING (AUTONOMOUS)

Accredited by NAAC with 'A' Grade & NBA (Under Tier - I), ISO 9001:2015 Certified Institution Approved by AICTE, New Delhi and Affiliated to JNTUK, Kakinada

L.B. REDDY NAGAR, MYLAVARAM, NTR DIST., A.P.-521 230.

Department of

COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the project entitled "**A Resnet-50 Model For The Detection Of Diabetic Retinopathy**" is being submitted by

Patiballa Surya Vamsi **19761A0547**

in partial fulfillment of the requirements for the award of degree of B**. Tech** in **Computer Science & Engineering** from **Jawaharlal Nehru Technological University Kakinada** is a record of bonafide work carried out by them at **Lakireddy Bali Reddy College of Engineering.**

The results embodied in this Project report have not been submitted to any other University or Institute for the award of any degree or diploma

PROJECT GUIDE
Mr. N V Naik

HEAD OF THE DEPARTMENT
Dr. D.Veeraiah

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We take great pleasure to express our deep sense of gratitude to our project guide **Mr. N V Naik,** Sr**.** Assistant Professor, for his valuable guidance during the course of our project work.

We would like to thank **Dr. D. Veeraiah**, Professor & Head of the Department of Computer Science & Engineering for his encouragement.

We would like to express our heart-felt thanks to **Dr K. Appa Rao,** Principal, Lakireddy Bali Reddy College of Engineering for providing all the facilities for our project.

Our utmost thanks to all the faculty members and Non-Teaching Staff of the Department of Computer Science & Engineering for their support throughout our project work.

Our Family Members and Friends receive our deepest gratitude and love for their support throughout our academic year.

Patiballa Surya Vamsi **19761A0547**

DECLARATION

We are here by declaring that the project entitled **"A Resnet-50 Model For The Detection Of Diabetic Retinopathy"** work done by us. We certify that the work contained in the report is original and has been done by us under the guidance of our supervisor. The work has not been submitted to any other institute in preparing for any degree or diploma. We have followed the guidelines provided by the institute in preparing the report. We have confirmed to the norms and guidelines given in the Ethical Code of Conduct of the Institute. Whenever we have used materials (data, theoretical analysis, figures and text) from other sources, we have given due credit to them by citing them in the text of the report and giving their details in the references. Further, we have taken permission from the copyright's owner of the sources, whenever necessary**.**

Signature of the student

Patiballa Surya Vamsi **19761A0547**

ABSTRACT

Among the most significant reasons for vision loss globally, especially amongst elderly is diabetic retinopathy(DR). The primary contributing factor to this condition in the retina is a consequence of containing high sugar levels in the blood vessels. Typically, the alert indications go unnoticed. Diabetic Retinopathy (DR) prognosis in its early stages is heavily reliant on screening. For Retinopathy(DR) detection and classification, the system uses the ResNet50 model, a Convolutional Neural Networks (CNN) model that has already undergone training and is dependent on deep learning. The system proposed enables a DR classification taking into account ordinary eyes(No DR), mild DR(NPDR), moderate DR(NPDR), severe DR(NPDR), and proliferative DR(PDR), which can help ophthalmologists make a preliminary choice. Results are generated for DR classification and AUC scores in terms of classification accuracy.

LIST OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

DR - Diabetic retinopathy

DL - Deep learning

CNN - Convolutional neural networks

DM - Diabetes mellitus

ADA - American Diabetes Association

AAORP -American Academy of Ophthalmology Retina Panel

MA - Microaneurysms

HM - Hemorrhages

EX - soft and hard exudates

NPRD - Non-Proliferative Diabetic Retinopathy

HOG - Histogram of Oriented Gradients

PSO - Particle swarm optimization

PCA - Principal component analysis

1. INTRODUCTION

1.1 Overview of The Project

Diabetes, a chronic disease affects various organs in the human body, including the retina. Diabetic retinopathy (DR) results from diabetes mellitus (DM). DM is the leading cause of blindness among a significant age group in Western countries. It is increasing in underdeveloped countries too. Patients with DM are much more susceptible to blindness than without DM. Progressive diabetic retinopathy and macular edema (clinically significant) can lead to severe vision loss. DR affects a large diabetic population in developed countries, it is a silent disease that only appears in its later stages where treatment is very difficult and, in some cases, impossible. In the case of DR, the blood vessels that help nourish the retina begin to leak fluid and blood onto the retina, resulting in visual features called lesions such as microaneurysms, hemorrhages, hard exudates, cotton wool spots, vessel area blood. It can only be treated effectively in its early stages and therefore its early detection is very important through regular screening. Automatic screening is highly necessary so that manual effort is reduced since the cost of this procedure is quite high. Studies have shown that the blinding complications of diabetes can be largely prevented medically, through glycemic and blood pressure control, as well as the early detection and prompt treatment of diabetic retinopathy with photocoagulation/ surgical techniques. Therefore, screening guidelines have been developed by national professional organizations such as the American Diabetes Association (ADA) and American Academy of Ophthalmology Retina Panel (AAORP). Adults and children over 10 years of age with type 1 diabetes should have an initial dilated examination by an ophthalmologist within 5 years of the onset of diabetes. Since people may already have type 2 diabetes before they are aware of symptoms, and up to 20% of patients with type 2 diabetes have retinopathy at the time of diagnosis, they should have an initial dilated exam with an ophthalmologist when diagnosed with diabetes.

Subsequent examinations should be annual or more frequent if retinopathy progresses. Pregnant women with pre-existing diabetes should have a dilated eye exam early in the first trimester of pregnancy, as pregnancy may potentiate the rapid progression of retinopathy. DR is detected by the appearance of different types of lesions on an image retina. These lesions are microaneurysms (MA), hemorrhages (HM), soft and hard exudates (EX).

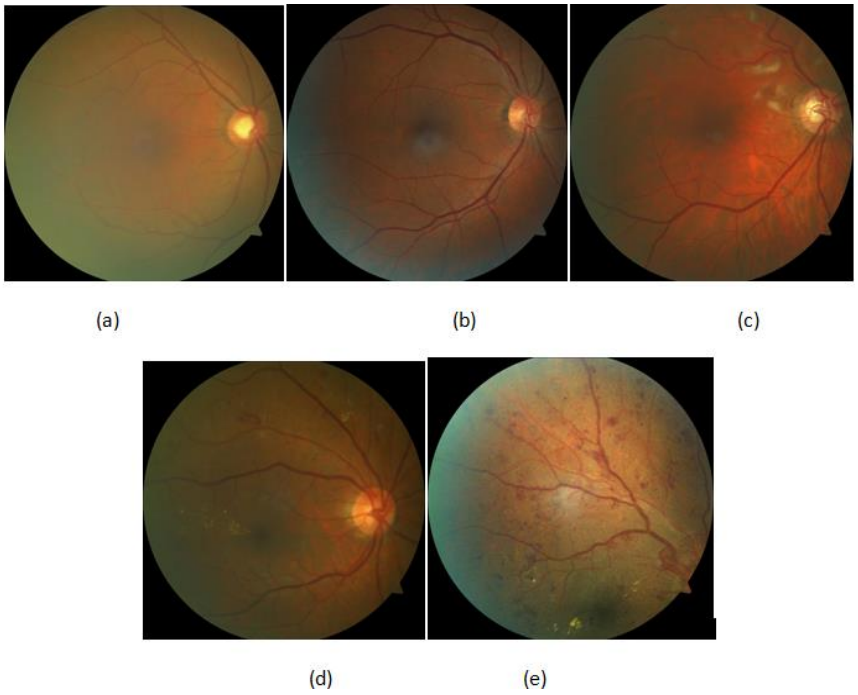

Figure-1: Classes of Retinal Images

a) Normal eyes

b) Mild DR

c) Moderate DR

d) Severe DR and

e) Proliferative DR

1.2 Feasibility Study

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

1.2.1 Economical Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

1.2.2 Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

1.2.3 Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar.

1.3 Scope

- To build a robust model to give correct and accurate results while the detection of diabetic retinopathy in any human eye.
- The present system is trained by ResNet-50 model which is a pre-trained model.
- As further enhancement, the work can be extended with more pre-trained model implementation such as AlexNet and VGG16. The work can be also extended with hyper parameter tuning to identify best fit parameter through Grid Search Cross Validation (GSCV) can be implemented in the future work.

2.LITERATURE SURVEY

Many existing techniques have been studied by the researchers on Diabetic retinopathy detection, few of them are discussed below.

- **Classification of Diabetic Retinopathy Stages using Histogram of Oriented Gradients and Shallow Learning**

  D. Sarwinda, T. Siswantining and A. Bustamam, 2018 International Conference on Computer, Control, Informatics and its Applications (IC3INA), 2018, pp. 83-87, doi: 10.1109/IC3INA.2018.8629502.

  This study classifies the stage of Diabetic Retinopathy (DR) into three classes, namely normal, mild Non-Proliferative Diabetic Retinopathy (NPDR), and moderate/severe NPDR class. In general, this research is done to solve that problem arises as a result of similarity of image per stages that cannot be assessed invisible. So, it requires a handling where the image of the retina can be categorized into appropriate categories. Based on the problem, two experimental mechanisms were conducted for each hierarchy, i.e approach computer vision that only focus to process the whole image and the approach taken by the medical using texture feature as marker feature to detect DR. The data are obtained from DiaretDB0 public database. The experimental result shows that our proposed method is able to provide good enough performance in terms of time and accuracy. This study achieved around 85% accuracy for the binary class classification.

  Techniques Used:

  The techniques used is Histogram of Oriented Gradients (HOG) to extract feature. To select the best feature from HOG, we used factor analysis as a feature selection method. This step was done to get good performance in classification step. In our experimental design, we implemented shallow learning such as Support Vector Machines learning and Random Forest learning to classify moderate/sever NPDR vs. mild NPDR, mild NPDR vs. Normal, and moderate/severe NPDR vs. Normal.

- **Detection of Early Signs of Diabetic Retinopathy Based on Textural and Morphological Information in Fundus Images**

  Colomer, Adrián et al., Sensors (Basel, Switzerland) 20 (2020)

  Estimated blind people in the world will exceed 40 million by 2025. To develop novel algorithms based on fundus image descriptors that allow the automatic classification of

retinal tissue into healthy and pathological in early stages is necessary. This study focus on one of the most common pathologies in the current society: diabetic retinopathy. The proposed method avoids the necessity of lesion segmentation or candidate map generation before the classification stage. Through several experiments, the ability of the proposed system to identify diabetic retinopathy signs is validated using different public databases with a large degree of variability and without image exclusion.

Techniques Used:

Local binary patterns and granulometric profiles are locally computed to extract texture and morphological information from retinal images. Different combinations of this information feed classification algorithms to optimally discriminate bright and dark lesions from healthy tissues.

- **Data Augmentation for Improving Proliferative Diabetic Retinopathy Detection in Eye Fundus Images**

  T. Araújo et al., in IEEE Access, vol. 8, pp. 182462-182474, 2020, doi: 10.1109/ACCESS.2020.3028960.

  Proliferative diabetic retinopathy (PDR) is an advanced diabetic retinopathy stage, characterized by neovascularization, which leads to ocular complications and severe vision loss. However, the available DR-labeled retinal image datasets have a small representation of images of the severest DR grades, and thus there is lack of PDR cases for training DR grading models. Additionally, the criteria for labelling these images in the publicly available datasets is not always clear, with some images which do not show typical PDR lesions being labeled as PDR due to the presence of photo-coagulation treatment and laser marks. This problem, together with the datasets' high class imbalance, leads to a limited variability of the samples, which the typical data augmentation and class balancing cannot fully mitigate. We propose a heuristic-based data augmentation scheme based on the synthesis of neovessel (NV)-like structures that compensates for the lack of PDR cases in DR-labeled datasets.

  Techniques Used:

  The proposed neo-vessel generation algorithm relies on the general knowledge of common location and shape of these structures. NVs are generated and introduced in pre-existent retinal images which can then be used for enlarging deep neural networks' training sets. The data augmentation scheme was tested on multiple datasets, and allows to improve the model's capacity to detect NVs.

- **Multi-Stream Deep Neural Network for Diabetic Retinopathy Severity Classification Under a Boosting Framework**

  H. Mustafa, S. F. Ali, M. Bilal and M. S. Hanif, in IEEE Access, vol. 10, pp. 113172-113183, 2022, doi: 10.1109/ACCESS.2022.3217216.

  Diabetic Retinopathy (DR) is an eye disorder in patients with diabetes. Detection of DR presence and its complications using fundus images at an early stage helps prevent its progression to the advanced levels. In the recent years, several well-designed Convolutional Neural Networks (CNN) have been proposed to detect the presence of DR with the help of publicly available datasets. However, these existing CNN-based classifiers focus on utilizing different architectural settings to improve the performance of detection task only i.e. presence or absence of DR. The further classification of the severity and type of the disease, however, remains a non-trivial task. To this end, we propose a multi-stream ensemble deep network to classify diabetic retinopathy severity. The proposed approach takes advantages of the deep networks and principal component analysis (PCA) to learn inter-class and intra-class variations from the raw image features. Ensemble machine learning classifiers are then applied to achieve high classification accuracy and robust performance on the obtained deep features. The proposed approach has been compared with multiple conventional CNN-based approaches on Messidor-2 (two categories) and EyePACS (two, five categories) datasets.

  Techniques Used:

  Deep learning model to serve as the main feature extractors. Further application of PCA reduces the dimensionality of features and effectively separates the variation space of inter-class and intra-class images. Finally, an ensemble machine learning classifier using AdaBoost and random forest algorithms is built to further improve classification accuracy.

- **Feature Selection of Diabetic Retinopathy Disease Using Particle Swarm Optimization and Neural Network**

  A. Herliana, T. Arifin, S. Susanti and A. B. Hikmah," 2018 6th International Conference on Cyber and IT Service Management (CITSM), 2018, pp. 1-4, doi: 10.1109/CITSM.2018.8674295.

  Currently, there are many medical experts who face difficulty in conducting early detection for diabetic retinopathy. This occurs because it is difficult to recognize the early symptoms of this disease. In order for this disease to be detected early, an accurate

classification method is required. Data mining concept is one alternative in conducting classification. The study result show that there is an increase in result by applying neural network based particle swarm optimization (PSO) of 76.11%. This study also show that there is an increase in classification result by using feature selection method of 4.35% from previous result of 71.76% by only applying neural network method.

Techniques Used:

This study was conducting by applying particle swarm optimization (PSO) method to select the best Diabetic Retinopathy feature based on diabetic retinopathy dataset. Then, the selected feature is further classified using classification method of neural network.

INFERENCE FROM THE LITERATURE SURVEY

The literature study that there are many researches based on Diabetic retinopathy detection was proposed. Some of the used the feature extraction techniques like particle swarm optimization (PSO) and feature reduction techniques such as PCA (principal component analysis). Though the features are extracted or reduced through efficient techniques, they are not given high classification accuracy. Thus automatic feature extraction by the deep learning model is preferred in our study. The ResNet 50 model based on Convolutional Neural Network has the ability to extract the features automatically from the images is proffered for this study.

2.1 Existing System & Drawbacks

The advancement of automated DR pathology screening during the last few decades has been encouraging. In the literature, many deep learning and machine learning-based techniques have been presented. Akram et al. detected the presence of lesions in the retina using a mixture ensemble classifier built on the Gaussian Mixture Model (GMM) and Support Vector Machine (SVM). By combining the shape enhanced feature set with the intensity features, a similar strategy has been utilized to improve the model's classification accuracy by Akram et al. Several classification techniques like k-Nearest Neighbors (k-NN), AdaBoost, SVM, and GMM were applied and their performances were evaluated to detect lesions from non-lesions in the provided retinal images. The area of hard exudates, the area of veins and arteries, branching points, texture, and entropy were extracted from retinal images using a hybrid feature extraction technique.

Limitations:

The techniques discussed above are not good in performance because they employ traditional classification techniques, which may not be enough for distinguishing between complicated actual data such as lesion and non-lesion pictures. Additionally, the domain knowledge of the input data is required by the approaches utilized in these feature engineering methods. Deep learning, particularly CNNs, offers significant assistance in addressing DR classification issues. Deep learning models can detect minor local characteristics straight from retinal pictures without the need for human assistance or domain expertise. Gulshan et al. used the Incepetionv3 for the diabetic retinopathy detection. The EyePACS-1 dataset, that includes 9963 images, as well as the Messidor-2 dataset were used to evaluate the model. According to their research the CNN-based models offers great sensitivity and specificity for diagnosing DR. Pratt et al. suggested a CNN that can classify the retinal images in the five stages of the DR and also detect the haemorrhages, micro-aneurysms, and exudates.

2.2 Proposed System

- The proposed system is diabetic retinopathy severity detection from retinal images.
- The proposed work uses pre-trained deep learning model based on Convolutional Neural Network (CNN) called ResNet50 model.
- The model is evaluated for accuracy and loss, AUC score.

Advantages

- The proposed system is cost effective.
- It is trained by Residual Network model that is 50 layers deep which has more predictive capabilities.
- The accuracy of the system increases.
- The time required for the detection of diabetic retinopathy is minimized compared to previous models.
- Thus the proposed system detects the Diabetic Retinopathy in less time with more accuracy.

3.SYSTEM ANALYSIS

3.1 Overview of System Analysis

3.1.1 Requisites Accumulating and Analysis

Computer Aided learning is a rapidly growing dynamic area of research in medical diagnosis system. The recent researchers in machine learning and Deep Learning promise the improved accuracy of perception of various diseases. Here the computers are enabled to think by developing intelligence by learning. There are many types of Machine Learning Techniques and which are used to classify the data sets.

Functional Requirements

The proposed application should be able to identify the type of DR class from 5 types of classes. Functional requirements define a function of a system and its components. A function is described as a set of inputs, the behavior and its outputs.

Functionality

The application is developed in such a way that any future enhancement can be easily implementable. The project is developed in such a way that it requires minimal maintenance. The software used are open source and easy to install. The application developed should be easy to install and use.

Reliability

It is the maturity, fault tolerance and recoverability. The system is reliable for any number of user input and training dataset. The proposed system used data augmentation for increasing the number of records in data.

Usability

It is easy to understand, learn and operate the software system.

Safety

Safety-critical is issues associated with its integrity level. The computer system being used is protected by a password.

Security

It does not block the some available ports through the Windows firewall.

Robustness

The application is developed in such a way that any future enhancement can be easily implementable. The project is developed in such a way that it requires minimal maintenance. The software used are open source and easy to install. The application developed should be easy to install and use.

Communications

The application is developed in such a way that it is easy to identify disease severity of Diabetic retinopathy images.

Non-Functional Requirements

Non-functional requirements determine the resources required, time interval, transaction rates, throughput and everything that deals with the performance of the system.

Maintainability

It is easy to maintain the system as it does not require any special maintenance after download. Updates are required only if notified to the user about any. Easy maintenance is one among the features that makes this proposal most usable.

Portability

The software must easily be transferred to another environment, including install ability. It is easily portable as it is implied on a regular computer. The user can access the computer from the place where the system was installed.

Performance

Less time for detection once the input is arrived. Similarly, the training time also less as we given limited epoch on training.

Accuracy

The accuracy generated by our work is outperformed than any other existing models. We can recognize the severity of disease accurately.

3.1.2 System Design

The system focusses on the training of the model using the given retina images dataset, which is done using pre-trained ResNet-50 model of Convolutional Neural Networks in Deep Learning.The training model includes both the sample output data and the communication of the input data set, which has a significant impact on the results. By using the proposed algorithm, the training model utilizes the input data to correlate the prepared output, as opposed to the sample output. The output from the correlation of the modified model is used to improve the accuracy of the predictions. In the training step, the iterative model is called "model fitting," where the model is fine-tuned to achieve better results. Preprocessing is an essential step in the system design, where the data is cleaned to remove noisy data containing hashes, special symbols, quality issues, and irrelevant and incomplete data. By implementing these steps in the system design, the project can create an efficient and accurate deep learning model for the detection of Diabetic Retinopathy.

3.1.3 Implementation

The implementation of the project is done by considering following steps:

- **Collection Of Dataset**

  The dataset used for this project is collected from open source Kaggle platform.

- **Data Preprocessing**

  Pre-processing data to ensure that it is in a format that the network can accept is a common first step in deep learning workflows. For example, you can resize image input to match the size of an image input layer. You can also preprocess data to enhance desired features or reduce artifacts that can bias the network.

- **Training of the model (ResNet-50)**

  Training is the most important step in deep learning. In training, you pass the prepared data to your deep learning model to find patterns and make predictions. It results in the model learning from the data so that it can accomplish the task set. Over time, with training, the model gets better at predicting.

- **Evaluating the Model**

  After training your model, you have to check to see how it’s performing. This is done by testing the performance of the model on previously unseen data. The unseen data used is the testing set that you split our data into earlier. If testing was done on the same data which is used for training, you will not get an accurate

measure, as the model is already used to the data, and finds the same patterns in it, as it previously did.

- **Making Predictions**

  In the end, the model to make predictions accurately on unseen data.

3.1.4 Testing

Quality Assurance:

Quality Assurance is popularly known as QA Testing, is defined as an activity to ensure that an organization is providing the best possible product or service to customers. QA focuses on improving the processes to deliver Quality Products to the customer. An organization has to ensure, that processes are efficient and effective as per the quality standards defined for software products.

Functional Test:

Functional Testing is also known as functional completeness testing, Functional Testing involves trying to think of any possible missing functions. As chat-bot evolves into new application areas, functional testing of essential chatbot components. Functional testing evaluates use-case scenarios and related business processes, such as the behavior of smart contracts.

3.1.5 Deployment of System

The deployment stage of the project involves the implementation of the trained model in a production environment. This typically involves incorporating the model into a software application or web service that can take in inputs such as retinal images of various categories for the detection of diabetic retinopathy. During deployment, it is important to ensure that the model is working correctly and accurately, and that any errors or discrepancies are identified and addressed promptly. Additionally, it may be necessary to continually monitor and update the model as new data becomes available, in order to maintain its accuracy and effectiveness over time.

3.2 Software Used in The Project

Python

Python is a translated, highly functional, common programming language. It is the language of many programs. The most widely used programming language with the highest level of the general-purpose program. Apart from being an open-source programming language, python is a targeted, translated and active programming language. Python combines remarkable power with clear syntax. It has modules, classes, variations, very powerful data types, and powerful typing. There are network connections for most system phones and libraries, as well as various windows installation programs. Python is also used as language extensions for applications written in other languages that require an easy-to-use script or interface. Python is widely regarded as the preferred language for teaching and learning DL (Deep Learning). Here are a few simple reasons:

- Easy to work with. Compared with C, C++, and Java, the syntax is simpler, and in Python, it is also comprised of a lot of code and libraries for ease-of-us.
- Although slower than other languages, the data management capacity is good.
- Open Source: – Python, along with the R, it is gaining momentum and popularity in the field of analysis, as both of these languages are open-source.
- Ability to communicate with almost all third-party languages and forums.

Packages

pandas

pandas software library written in the language of the Python program for cheating and analyzing data. In particular, it provides data structure and function for managing numerical tables and time series. pandas are widely used for machine learning in the form of data frames. Pandas allows importing data for various file formats such as csv, excel etc.

numpy

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

matplotlib

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK.

sklearn

scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support-vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy. Scikit-learn is a NumFOCUS fiscally sponsored project.

seaborn

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

SYSTEM REQUIREMENTS

3.2.1 Hardware Requirements

Processor	: Any Processor above 500 MHz.
Ram	: 4 GB
Hard Disk	: 250 GB
Input device	: Standard Keyboard and Mouse, Web Camera
Output device	: High Resolution Monitor.

3.2.2 Software Requirements

Operating System	: Windows 7 or higher
Programming	: Python 3.6 and related libraries

4. SYSTEM DESIGN

4.1 Overview of System Design

System Design

This chapter gives overview of architecture design, dataset for implementation, algorithm used and UML designs.

SYSTEM ARCHITECTURE

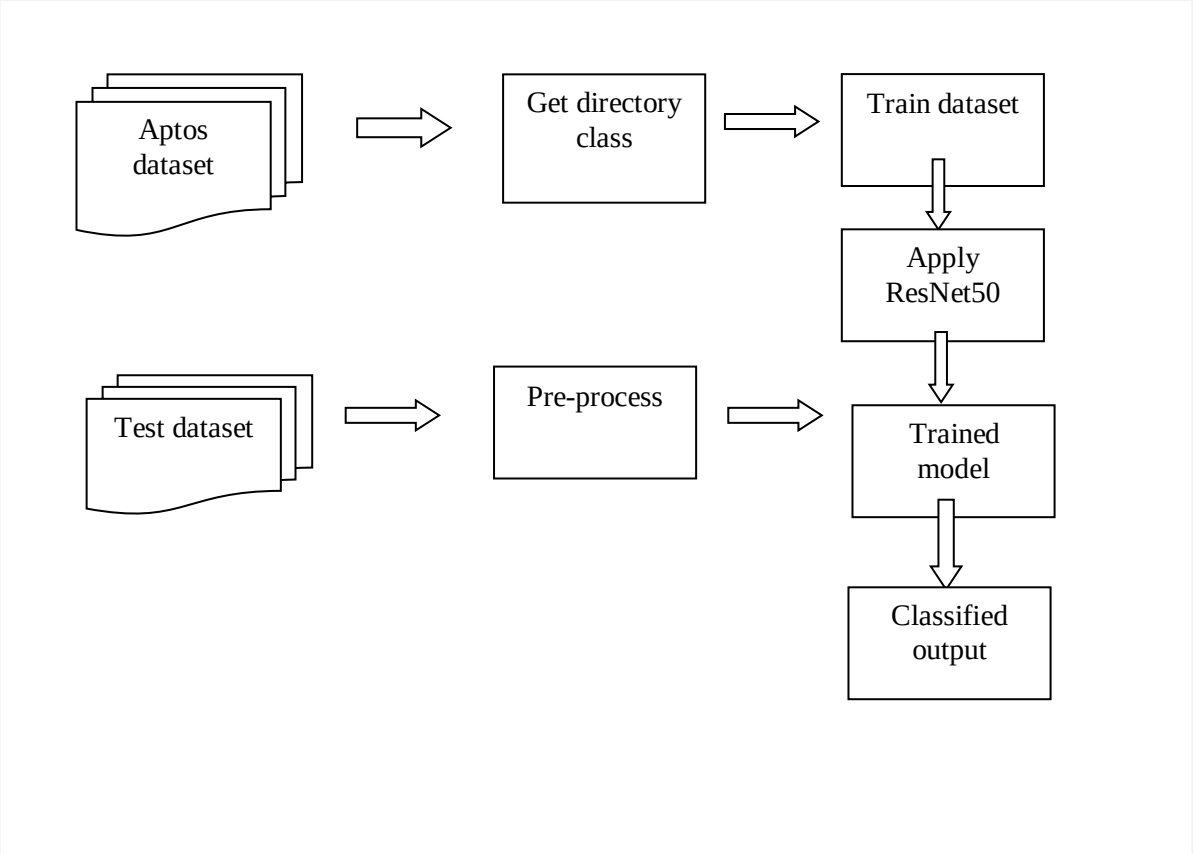

Figure-2: System Architecture

The above figure represents system architecture of the proposed system. First the dataset is loaded, pre-processed and split into train and test sets then, we apply Deep learning algorithms ResNet50. The predicted results are evaluated and performance of the algorithm is computed.

UML DIAGRAMS

The design is a plan or drawing produced to show the look and function or workings of an object before it is made. Unified Modeling language (UML) is a standardized modeling language enabling developers to specify, visualize, construct and document artifacts of a software system. Thus, UML makes these artifacts scalable, secure and robust in execution. UML is an important aspect involved in object-oriented software development. It uses graphic notation to create visual models of software systems.

The different types of UML diagram are as follows.

- Use Case Diagram
- Class Diagram
- Activity Diagram
- Sequence Diagram
- Collaboration Diagram
- Component Diagram
- Deployment Diagram

The following are the multiple kinds of Uml diagrams that were used in our project. We have used

- Use Case Diagram
- Sequence Diagram
- Collaboration Diagram
- Activity Diagram
- Deployment Diagram

USE CASE DIAGRAM

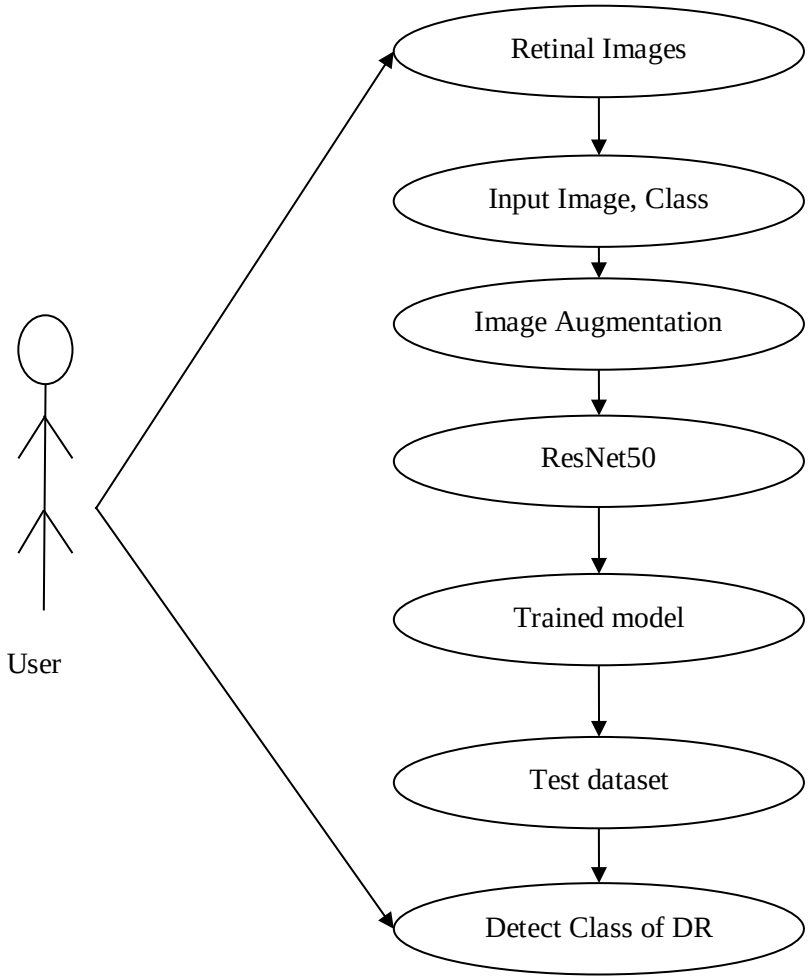

Figure-3: Use case Diagram

The above figure represent use case diagram of proposed system, where user inputs image of retinal images, the algorithm work to generate the identified output as type of DR. The actor and use case is represented. An eclipse shape represents the use case namely input image, pre-process, training, recognition and output.

SEQUENCE DIAGRAM

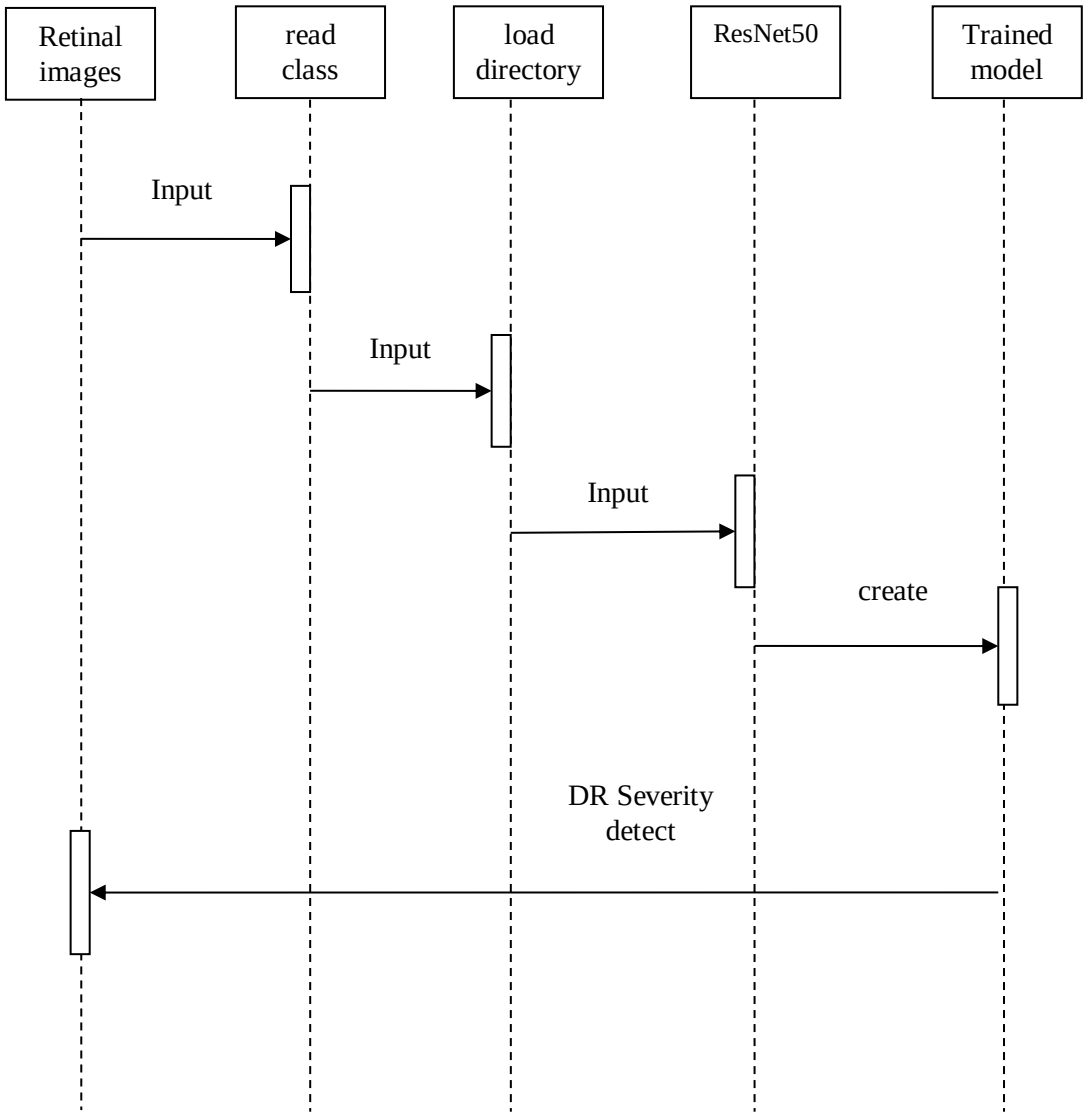

Figure-4: Sequence Diagram

A sequence diagram shows a parallel vertical lines, different processes or objects that live simultaneously, and as horizontal arrows, the messages exchanged between them, in order in which they occur. The above figure represents sequence diagram, the proposed system's sequence of data flow is represented.

COLLABORATION DIAGRAM

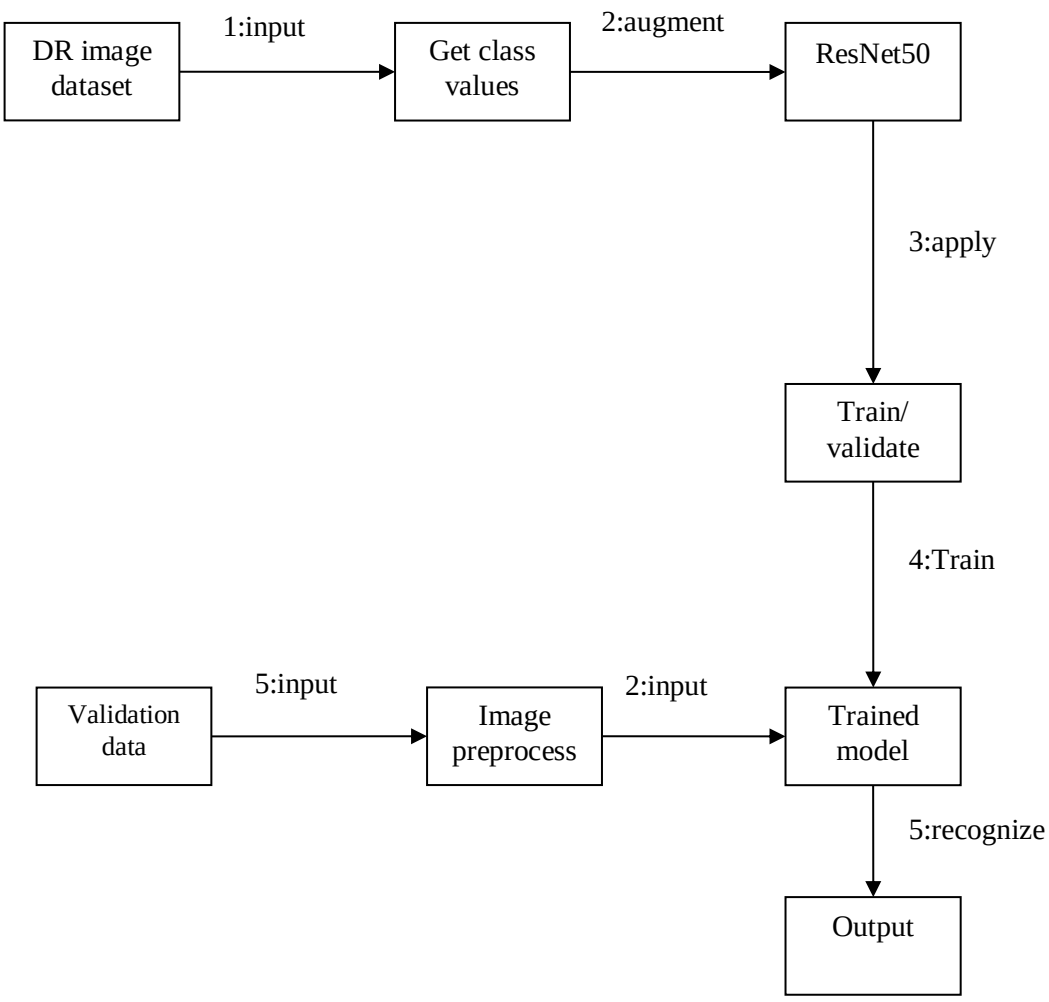

Figure-5: Collaboration Diagram

The above figure shows the collaboration diagram of the proposed system, where we represented the collaboration between the actor and function modules with sequence number.

ACTIVITY DIAGRAM

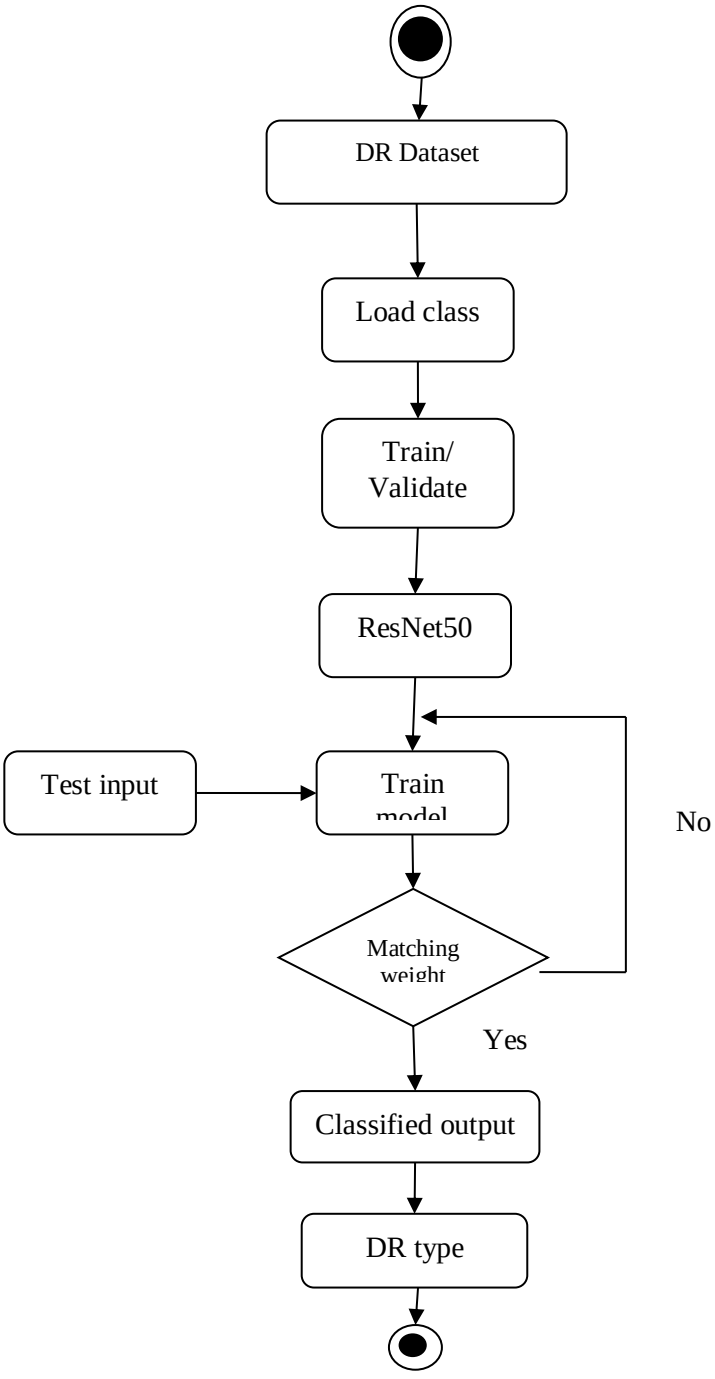

Figure-6: Activity Diagram

The above figure show the activity diagram of the proposed system, where we represented the identified activities and its functional flow.

DEPLOYMENT DIAGRAM

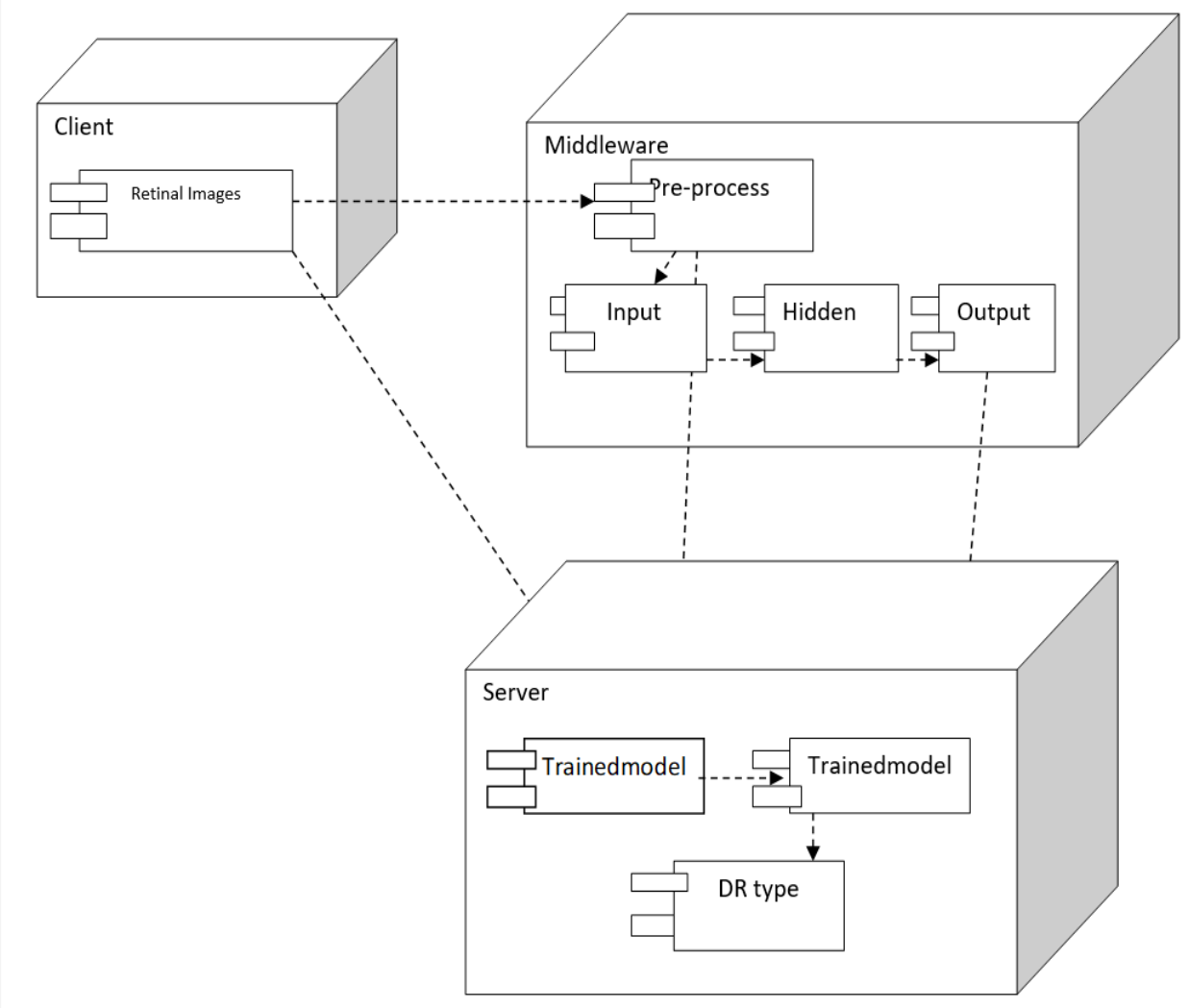

Figure-7: Deployment Diagram

In the deployment diagram the UML models the physical deployment of artifacts on nodes. The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Nodes may have sub-nodes, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers.

5.CODING & IMPLEMENTATION

IMPLEMENTATION

MODULES

The modules included in our implementation are as follows

- Dataset collection
- Image pre-processing
- ResNet50 model building

DATASET DESCRIPTION

Retinal fundus images are captured by an optical system known as a background camera, a combination of camera attached with a low-power microscope. It can be used simultaneously to illuminate the retina and its imaging. It was designed to image the inside of the eye primarily the retina, optic disc, macula and posterior pole. The present works consists of the DR classification using the Diabetic Retinopathy Detection 2015 challenge dataset available in https://www.kaggle.com/c/aptos2019-blindness-detection. There is a large set of high-resolution retinal images taken under various imaging conditions. A left and right field is provided for each subject. Images are tagged with a subject id as well as either left or right (i.e 1_left.png is the left eye of patient id 1). The dataset consists of 5593 files of 10.22 GB and consists of 35126 retinal images to detect diabetic retinopathy and all images are resized into 224x224 pixels. A clinician rated the presence of diabetic retinopathy in each image on a scale of 0 to 4, according to the scale mentioned previously.

- 0 - No DR
- 1 - Mild
- 2 - Moderate
- 3 - Severe
- 4 - Proliferative DR

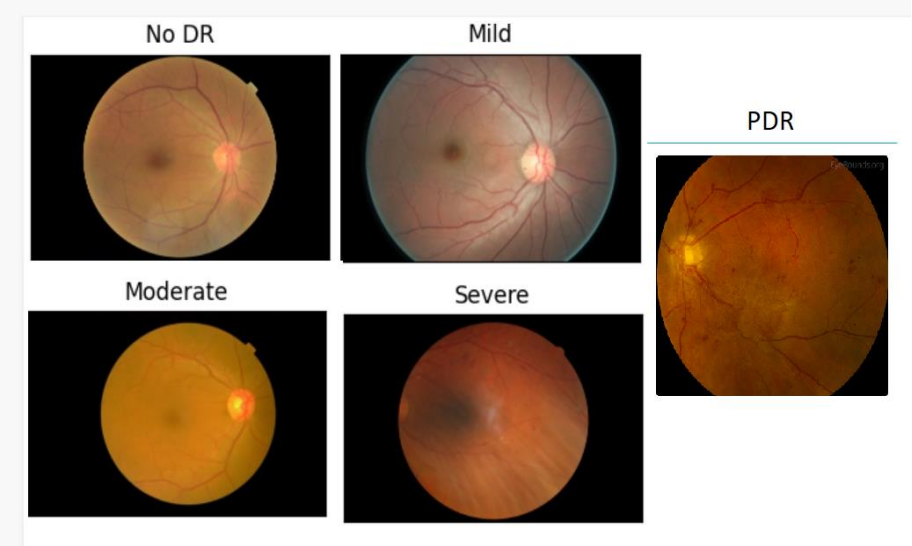

Figure-8: Classification of dataset (Scale 0-4)

❖ **The below is the snippet of our dataset used**

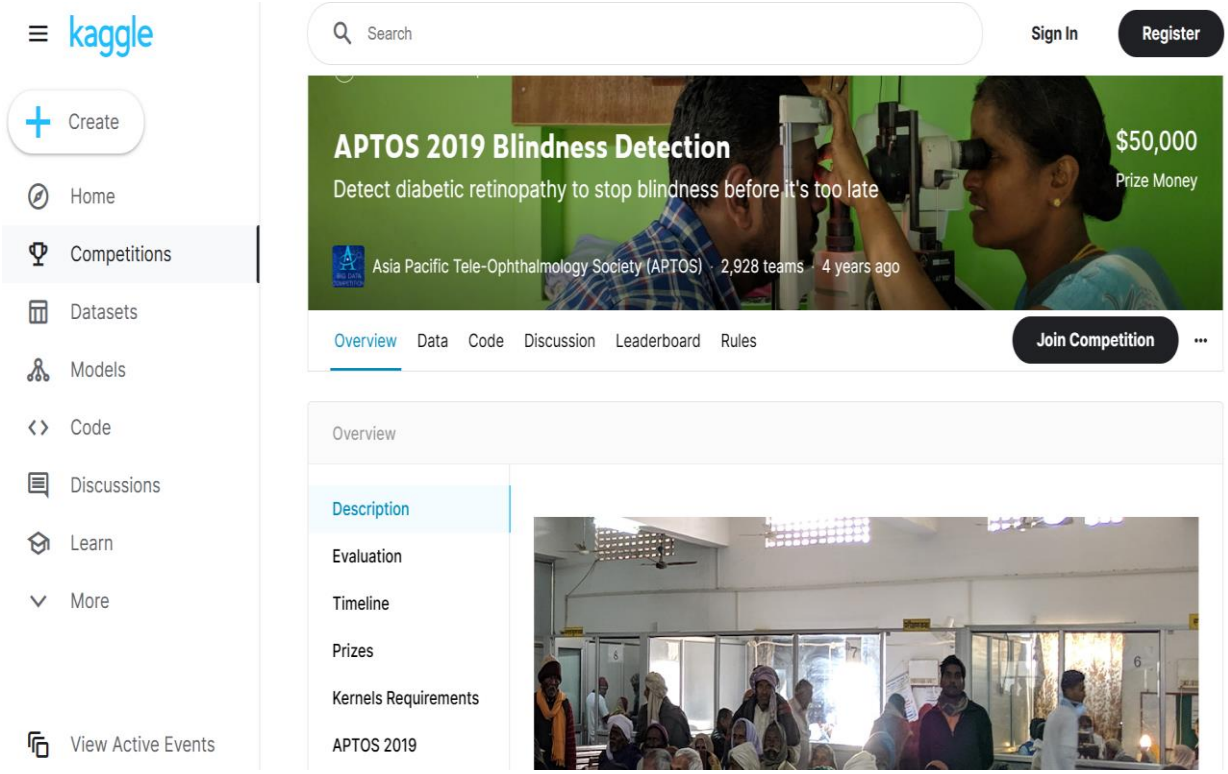

Figure-9: Snippet of dataset

❖ **The following screen shows the number of images in each class of DR dataset**

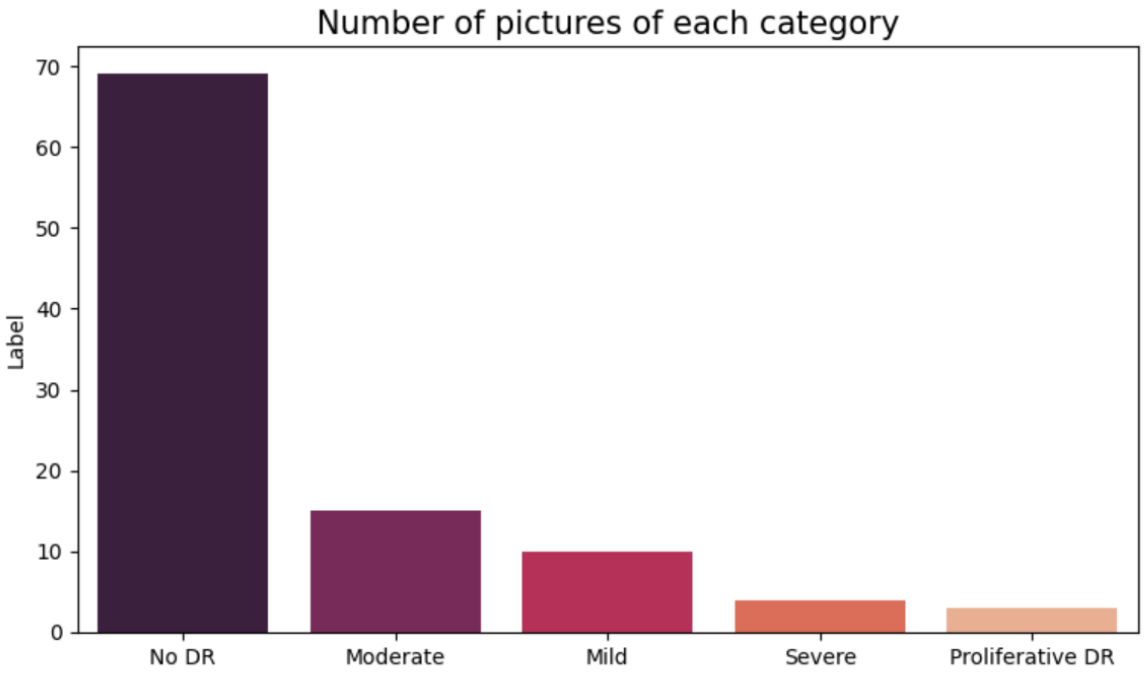

Figure-10: Number of pictures in each category

❖ **The following screen represents the csv file of the dataset.**

A	B	C	D
image	level		
10_left	No DR		
10_right	No DR		
13_left	No DR		
13_right	No DR		
15_left	Mild		
15_right	Moderate		
16_left	Proliferative DR		
16_right	Proliferative DR		
17_left	No DR		
17_right	Mild		
19_left	No DR		
19_right	No DR		
20_left	No DR		
20_right	No DR		
21_left	No DR		
21_right	No DR		
22_left	No DR		
22_right	No DR		

Figure-11: Screen of TrainLabels.csv file

IMAGE PRE-PROCESSING

There are few images pre-processing steps handled to make the data suitable for learning. Image Augmentation is a data augmentation method that generates more data from the existing samples. Keras ImageDataGenerator class provides way to augment images from the dataset. It provides a host of different augmentation techniques like standardization, rotation etc. In our pre-process step, data augmentation for Training, Validation and Test dataset is done.

ResNet-50 MODEL

ResNet-50 is a convolutional neural network that is 50 layers deep. This problem of training very deep networks has been alleviated with the introduction of ResNet or residual networks and these Resnets are made up from Residual Blocks. The ResNet architecture follows two basic design rules. First, the number of filters in each layer is the same depending on the size of the output feature map. Second, if the feature map's size is halved, it has double the number of filters to maintain the time complexity of each layer.

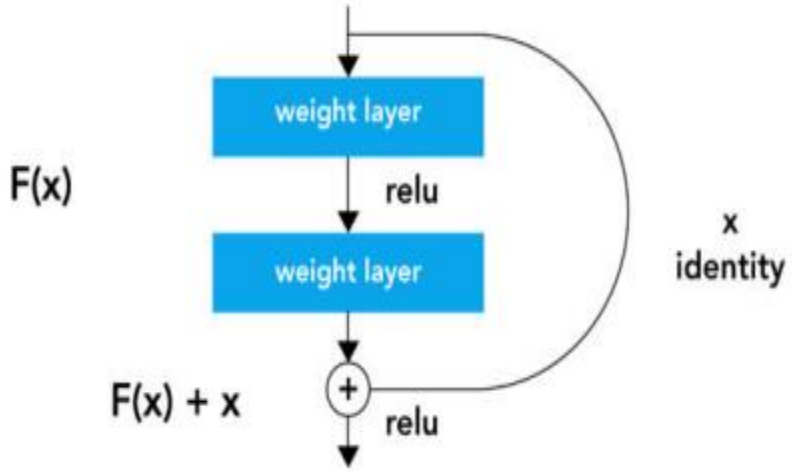

Figure-12: Residual Learning block

There is a direct connection which skips some layers(may vary in different models) in between. This connection is called 'skip connection' and is the core of residual blocks. Due to this skip connection, the output of the layer is not the same now. Without using this skip

connection, the input ‘x’ gets multiplied by the weights of the layer followed by adding a bias term.

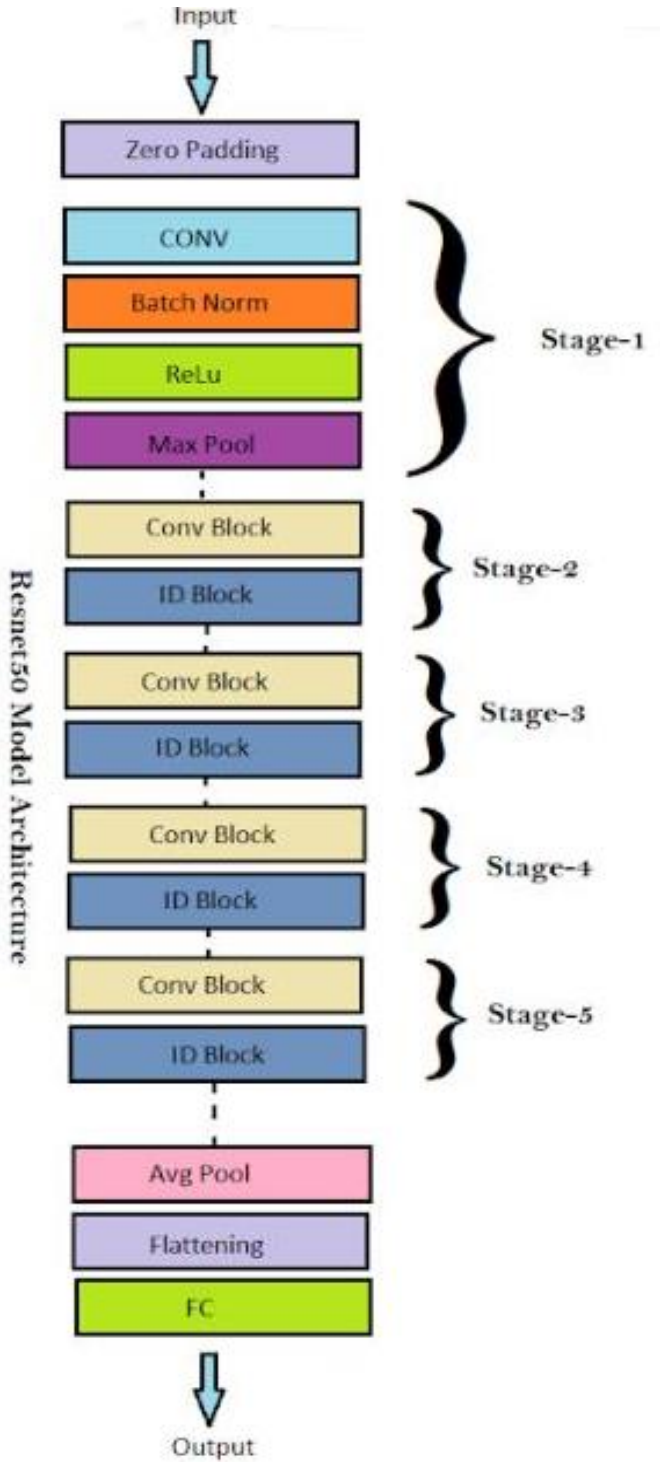

Figure-13: ResNet-50 Model Architecture

The 50-layer ResNet architecture includes the following elements are A 7×7 kernel convolution alongside 64 other kernels with a 2-sized stride. A max pooling layer with a 2-sized stride. 9 more layers—3×3,64 kernel convolution, another with 1×1,64 kernels, and a third with 1×1,256 kernels. These 3 layers are repeated 3 times. 12 more layers with 1×1,128 kernels, 3×3,128 kernels, and 1×1,512 kernels, iterated 4 times. 18 more layers with 1×1,256 cores, and 2 cores 3×3,256 and 1×1,1024, iterated 6 times. 9 more layers with 1×1,512 cores, 3×3,512 cores, and 1×1,2048 cores iterated 3 times. (up to this point the network has 50 layers). Average pooling, followed by a fully connected layer with 1000 nodes, using the softmax activation function.

Arguments used by ResNet50

- **include_top**: whether to include the fully-connected layer at the top of the network.
- **weights**: one of None (random initialization), 'Imagenet' (pre-training on ImageNet), or the path to the weights le to be loaded.
- **input_tensor**: optional Keras tensor (i.e. output of layers.Input()) to use as image input for the model.
- **input_shape**: optional shape tuple, only to be specied if include_top is False (otherwise the input shape has to be (224, 224, 3) (with 'channels_last' data format) or (3, 224, 224) (with 'channels_rst' data format). It should have exactly 3 inputs channels, and width and height should be no smaller than 32. E.g. (200, 200, 3) would be one valid value.
- **pooling**: Optional pooling mode for feature extraction when include_top is False.
- None means that the output of the model will be the 4D tensor output of the last convolutional block.
- avg means that global average pooling will be applied to the output of the last
- convolutional block, and thus the output of the model will be a 2D tensor.
- max means that global max pooling will be applied.
- **classes**: optional number of classes to classify images into, only to be specied if include_top is True, and if no weights argument is specified.

SOURCE CODE

```
from cgi import print_directory
import numpy as np
import pandas as pd
from pathlib import Path
import os.path
import matplotlib.pyplot as plt
from IPython.display import Image, display, Markdown
import matplotlib.cm as cm
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
import tensorflow as tf
from time import perf_counter
import seaborn as sns

trainLabels = pd.read_csv("trainLabels.csv")
print(trainLabels.head())

listing = os.listdir("Dataset/")
#listing.remove("trainLabels.csv")
print(np.size(listing))
# input image dimensions
img_rows, img_cols = 200, 200

filepaths = []
labels = []

i=0
for file in listing:
    print(file)
    base = os.path.basename("Dataset/" + file)
```

```
    fileName = os.path.splitext(base)[0]
    filepaths.append("Dataset/" + file)
    labels.append(trainLabels.loc[trainLabels.image==fileName, 'level'].values[0])
    i=i+1
    print(i)
    if i>100:
        break
print(filepaths)
print(labels)
filepaths = pd.Series(filepaths, name='Filepath').astype(str)
labels = pd.Series(labels, name='Label')

# Concatenate filepaths and labels:
image_df = pd.concat([filepaths, labels], axis=1)

# Shuffle the DataFrame and reset index:
image_df = image_df.sample(frac=1).reset_index(drop = True)

# Show the result:
print(image_df.head(3))



# Display some pictures of the dataset with their labels:
fig, axes = plt.subplots(nrows=3, ncols=4, figsize=(10, 7),
                         subplot_kw={'xticks': [], 'yticks': []})

for i, ax in enumerate(axes.flat):
    ax.imshow(plt.imread(image_df.Filepath[i]))
    ax.set_title(image_df.Label[i])
plt.tight_layout()
plt.show()
```

```
# Display the number of pictures of each category:
vc = image_df['Label'].value_counts()
plt.figure(figsize=(9,5))
sns.barplot(x = vc.index, y = vc, palette = "rocket")
plt.title("Number of pictures of each category", fontsize = 15)
plt.show()


def create_gen():
    # Load the Images with a generator and Data Augmentation:
    train_generator = tf.keras.preprocessing.image.ImageDataGenerator(
        preprocessing_function=tf.keras.applications.mobilenet_v2.preprocess_input,
        validation_split=0.1
    )

    test_generator = tf.keras.preprocessing.image.ImageDataGenerator(
        preprocessing_function=tf.keras.applications.mobilenet_v2.preprocess_input
    )

    train_images = train_generator.flow_from_dataframe(
        dataframe=train_df,
        x_col='Filepath',
        y_col='Label',
        target_size=(224, 224),
        color_mode='rgb',
        class_mode='categorical',
        batch_size=32,
        shuffle=True,
        seed=0,
        subset='training',
        rotation_range=30, # Uncomment to use data augmentation!
        zoom_range=0.15,
        width_shift_range=0.2,
```

```
    height_shift_range=0.2,
    shear_range=0.15,
    horizontal_flip=True,
    fill_mode="nearest"
)

val_images = train_generator.flow_from_dataframe(
    dataframe=train_df,
    x_col='Filepath',
    y_col='Label',
    target_size=(224, 224),
    color_mode='rgb',
    class_mode='categorical',
    batch_size=32,
    shuffle=True,
    seed=0,
    subset='validation',
    rotation_range=30, # Uncomment to use data augmentation!
    zoom_range=0.15,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.15,
    horizontal_flip=True,
    fill_mode="nearest"
)

test_images = test_generator.flow_from_dataframe(
    dataframe=test_df,
    x_col='Filepath',
    y_col='Label',
    target_size=(224, 224),
    color_mode='rgb',
    class_mode='categorical',
```

```
        batch_size=32,
        shuffle=False
    )

    return train_generator,test_generator,train_images,val_images,test_images



def get_model(model):
# Load the pretained model:
    kwargs =    {'input_shape':(224, 224, 3),
                 'include_top':False,
                 'weights':'imagenet',
                 'pooling':'avg'}

    pretrained_model = model(**kwargs)
    pretrained_model.trainable = False

    inputs = pretrained_model.input

    x = tf.keras.layers.Dense(128, activation='relu')(pretrained_model.output)
    x = tf.keras.layers.Dense(128, activation='relu')(x)

    outputs = tf.keras.layers.Dense(5, activation='softmax')(x)

    model = tf.keras.Model(inputs=inputs, outputs=outputs)

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

    return model
```

```
train_df, test_df = train_test_split(image_df, train_size=0.9, shuffle=True, random_state=1)
# Create the generators:

train_generator,test_generator,train_images,val_images,test_images=create_gen()

# Load the pretained model:

pretrained_model = tf.keras.applications.ResNet50(
    input_shape=(224, 224, 3),
    include_top=False,
    weights='imagenet',
    pooling='avg'
)

pretrained_model.trainable = False


inputs = pretrained_model.input

x = tf.keras.layers.Dense(128, activation='relu')(pretrained_model.output)
x = tf.keras.layers.Dense(128, activation='relu')(x)

outputs = tf.keras.layers.Dense(5, activation='softmax')(x)

model = tf.keras.Model(inputs=inputs, outputs=outputs)

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

```
)
history = model.fit(
    train_images,
    validation_data=val_images,
    batch_size = 32,
    epochs=100)

pd.DataFrame(history.history)[['accuracy','val_accuracy']].plot()
plt.title("Accuracy")
plt.show()

pd.DataFrame(history.history)[['loss','val_loss']].plot()
plt.title("Loss")
plt.show()

pd.DataFrame(history.history)[['auc','val_auc']].plot()
plt.title("auc")
plt.show()

#results = model.evaluate(test_images, verbose=0)

#print_directory(" ## Test Loss: {:.5f}".format(results[0]))
#print_directory("## Accuracy on the test set: {:.2f}%".format(results[1] * 100))
#print('\n')


pred = model.predict(test_images)
pred = np.argmax(pred,axis=1)
print(pred)
y_test = list(test_df.Label)
print(y_test)
```

6.SYSTEM TESTING

6.1 Overview of Testing

Software testing is a process that will be used to evaluate software quality. Software testing is a powerful technology test designed to provide participants with information about the quality of a tested product or service, depending on the context in which it is intended to operate. This information may include, but is not limited to, the application process or application for tracking errors. Quality is not absolute, it is the value of one person. It is for this reason that tests may not be fully detected by the accuracy of any software, and criticism of tests or comparisons that compare the status and behavior of the product is kept to a minimum. It is very important to ensure that software testing should be separated from a separate discipline, Software Quality Assurance (S. Q. A.), which covers all areas of business processes, not just tests.

6.2 Types of Tests

Software testing methods are basically classified into BLACK BOX TESTING and WHITE-BOX TESTING. These two methods can be used to explain the point that a test engineer takes when developing the test cases.

6.2.1 Black Box Testing

The black box test works with software as a black box without understanding internal behavior. It aims to test performance. Therefore, the tester enters the data and sees only the output of the test object. This standard of testing often requires that the full test cases provided to the inspector can simply confirm that of the given input, the amount of output (or character), is the same as the expected value stated in the test case. Methods of checking black boxes include: stock sorting, boundary analysis, all double checking, fuzz testing, model-based testing, matric tracking etc.

6.2.2 White Box Testing

The white box test, however, is when the tester achieves internal data structure, code and algorithms. White box check methods include creating tests to satisfy other coding methods. For example, a test designer can create a test to create that all statements in the system are made at least once. Some examples of white box tests are dating methods and error injection changes. The white box test includes all specific tests.

Testing can be done on the following levels

6.2.3 Unit Testing:

Unit test tests for minimum component or module. Each unit (basic component) of the software has been tested to ensure that the detailed structure of the block is performed correctly. In an object-focused environment, this is usually done in the classroom, and testing of small units involves builders and destroyers.

6.2.4 Integration Testing

Integration testing reveals defects in interface and interactions between integrated components (modules). Largely continuous groups of tested software components related to building materials are compiled and tested till the software works as a system.

6.2.5 System Testing

System testing tests a fully integrated system to ensure it meets its requirements. System integration testing ensures that the system is integrated into any external or third-party programs defined in system requirements.

7.RESULTS

ResNet50, a convolutional neural network architecture with a dataset of images representing 5 classes of diabetic retinopathy, is the platform used for the experiment. Python programming is used for the implementation, along with necessary libraries like Keras and Tensorflow. To boost the quantity of visuals, augmentation of data is used. The experiment is conducted up to 100 epochs and the experimental findings indicate that our model achieves a maximum validation AUC score of 94% and training AUC of 97.77% approximately.

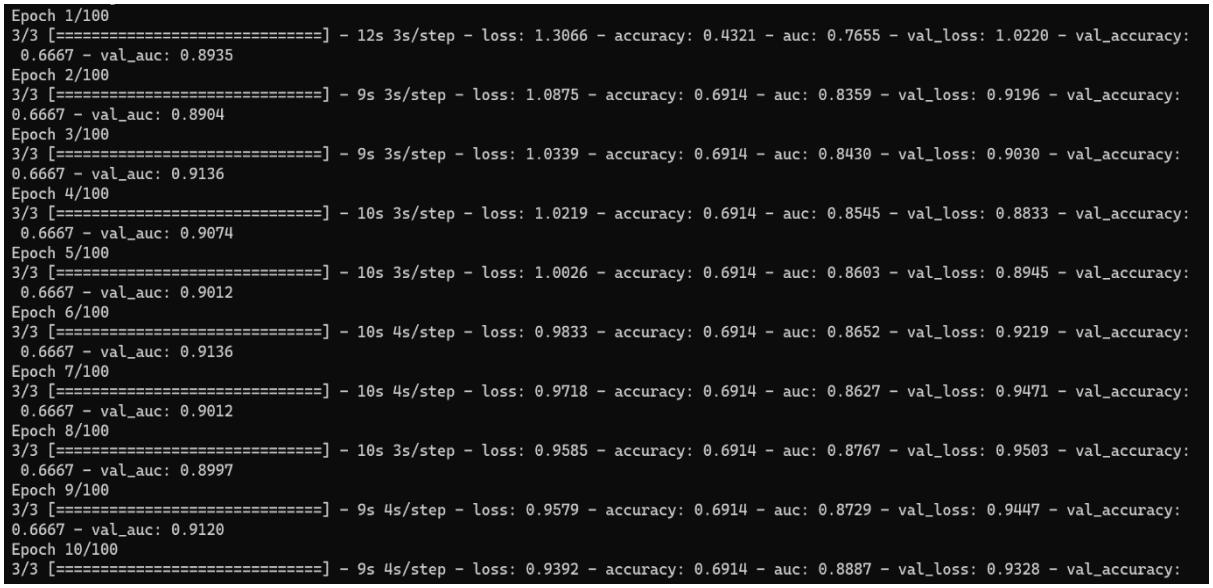

Figure-14: Result Screen (1-10 epochs)

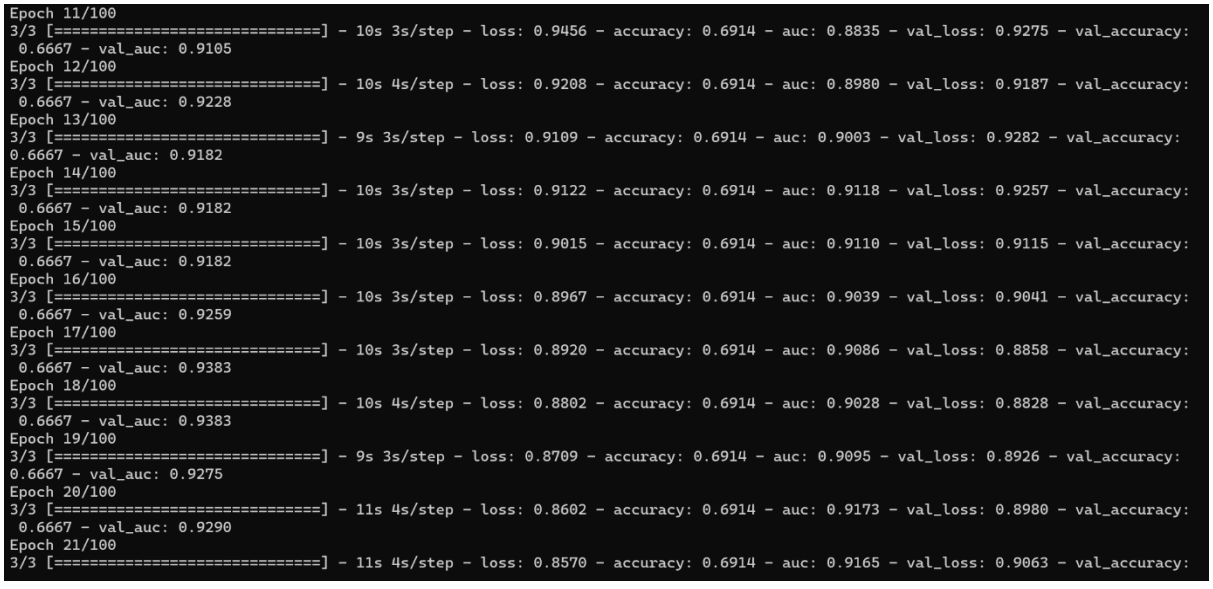

Figure-15 : Result Screen (11-21 epochs)

```
Epoch 21/100
3/3 [==============================] - 11s 4s/step - loss: 0.8570 - accuracy: 0.6914 - auc: 0.9165 - val_loss: 0.9063 - val_accuracy:
 0.6667 - val_auc: 0.9275
Epoch 22/100
3/3 [==============================] - 10s 3s/step - loss: 0.8517 - accuracy: 0.6914 - auc: 0.9116 - val_loss: 0.8747 - val_accuracy:
 0.6667 - val_auc: 0.9244
Epoch 23/100
3/3 [==============================] - 10s 4s/step - loss: 0.8492 - accuracy: 0.6914 - auc: 0.9120 - val_loss: 0.8497 - val_accuracy:
 0.6667 - val_auc: 0.9290
Epoch 24/100
3/3 [==============================] - 10s 3s/step - loss: 0.8373 - accuracy: 0.6914 - auc: 0.9158 - val_loss: 0.8595 - val_accuracy:
 0.6667 - val_auc: 0.9213
Epoch 25/100
3/3 [==============================] - 10s 4s/step - loss: 0.8247 - accuracy: 0.6914 - auc: 0.9165 - val_loss: 0.8669 - val_accuracy:
 0.6667 - val_auc: 0.9259
Epoch 26/100
3/3 [==============================] - 10s 3s/step - loss: 0.8224 - accuracy: 0.6914 - auc: 0.9164 - val_loss: 0.8865 - val_accuracy:
 0.6667 - val_auc: 0.9213
Epoch 27/100
3/3 [==============================] - 10s 3s/step - loss: 0.8170 - accuracy: 0.6914 - auc: 0.9175 - val_loss: 0.9027 - val_accuracy:
 0.6667 - val_auc: 0.9167
Epoch 28/100
3/3 [==============================] - 10s 3s/step - loss: 0.8030 - accuracy: 0.7037 - auc: 0.9208 - val_loss: 0.8840 - val_accuracy:
 0.6667 - val_auc: 0.9244
Epoch 29/100
3/3 [==============================] - 11s 4s/step - loss: 0.7901 - accuracy: 0.7160 - auc: 0.9250 - val_loss: 0.8692 - val_accuracy:
 0.6667 - val_auc: 0.9198
Epoch 30/100
3/3 [==============================] - 11s 3s/step - loss: 0.7996 - accuracy: 0.7160 - auc: 0.9190 - val_loss: 0.8636 - val_accuracy:
 0.6667 - val_auc: 0.9228
Epoch 31/100
3/3 [==============================] - 12s 4s/step - loss: 0.7768 - accuracy: 0.7160 - auc: 0.9251 - val_loss: 0.8585 - val_accuracy:
```

Figure-16: Result Screen (21-31 epochs)

```
3/3 [==============================] - 12s 4s/step - loss: 0.7768 - accuracy: 0.7160 - auc: 0.9251 - val_loss: 0.8585 - val_accuracy:
 0.6667 - val_auc: 0.9275
Epoch 32/100
3/3 [==============================] - 12s 4s/step - loss: 0.8040 - accuracy: 0.7284 - auc: 0.9245 - val_loss: 0.8556 - val_accuracy:
 0.6667 - val_auc: 0.9275
Epoch 33/100
3/3 [==============================] - 11s 3s/step - loss: 0.7667 - accuracy: 0.7407 - auc: 0.9256 - val_loss: 0.8974 - val_accuracy:
 0.6667 - val_auc: 0.9167
Epoch 34/100
3/3 [==============================] - 10s 3s/step - loss: 0.7903 - accuracy: 0.6914 - auc: 0.9259 - val_loss: 0.9053 - val_accuracy:
 0.6667 - val_auc: 0.9151
Epoch 35/100
3/3 [==============================] - 11s 3s/step - loss: 0.7906 - accuracy: 0.7284 - auc: 0.9224 - val_loss: 0.8980 - val_accuracy:
 0.7778 - val_auc: 0.9290
Epoch 36/100
3/3 [==============================] - 11s 4s/step - loss: 0.7614 - accuracy: 0.7531 - auc: 0.9293 - val_loss: 0.8746 - val_accuracy:
 0.6667 - val_auc: 0.9198
Epoch 37/100
3/3 [==============================] - 11s 3s/step - loss: 0.7625 - accuracy: 0.7284 - auc: 0.9285 - val_loss: 0.8727 - val_accuracy:
 0.6667 - val_auc: 0.9198
Epoch 38/100
3/3 [==============================] - 12s 4s/step - loss: 0.7572 - accuracy: 0.7531 - auc: 0.9258 - val_loss: 0.8751 - val_accuracy:
 0.6667 - val_auc: 0.9198
Epoch 39/100
3/3 [==============================] - 13s 4s/step - loss: 0.7434 - accuracy: 0.7531 - auc: 0.9380 - val_loss: 0.9074 - val_accuracy:
 0.6667 - val_auc: 0.9136
Epoch 40/100
3/3 [==============================] - 12s 4s/step - loss: 0.7353 - accuracy: 0.7407 - auc: 0.9311 - val_loss: 0.9456 - val_accuracy:
 0.6667 - val_auc: 0.8951
Epoch 41/100
3/3 [==============================] - 12s 4s/step - loss: 0.7529 - accuracy: 0.7160 - auc: 0.9330 - val_loss: 0.8726 - val_accuracy:
 0.6667 - val_auc: 0.9198
```

Figure-17 : Result Screen (31-41 epochs)

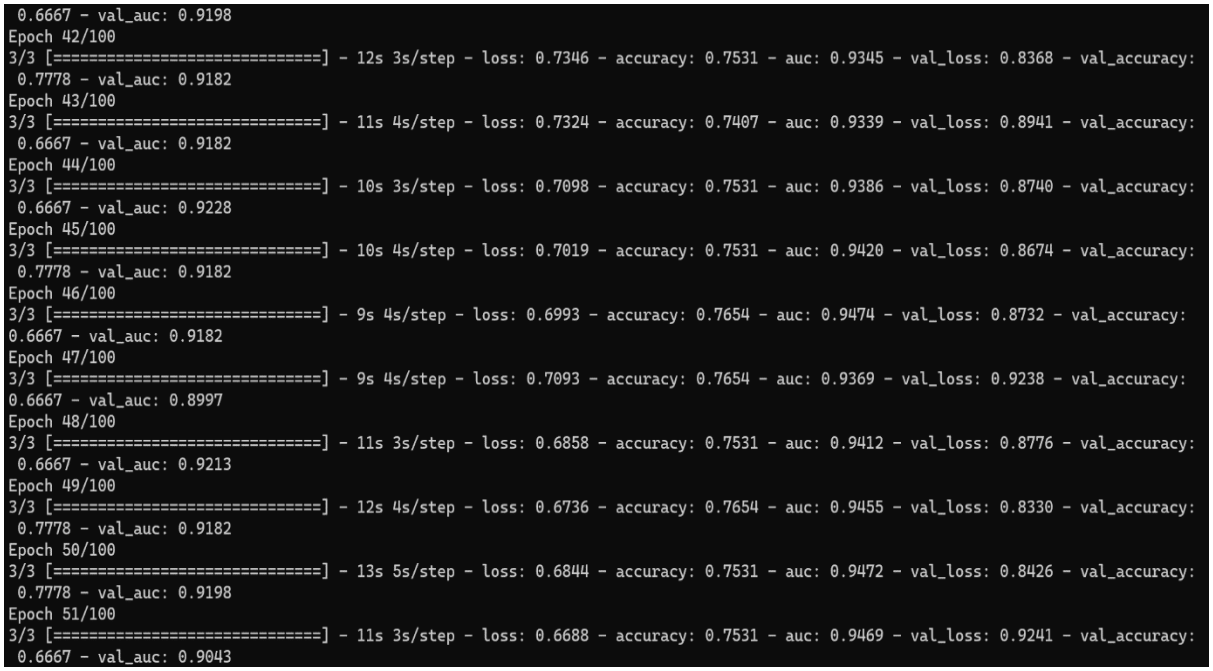

Figure-18 : Result Screen (42-51 epochs)

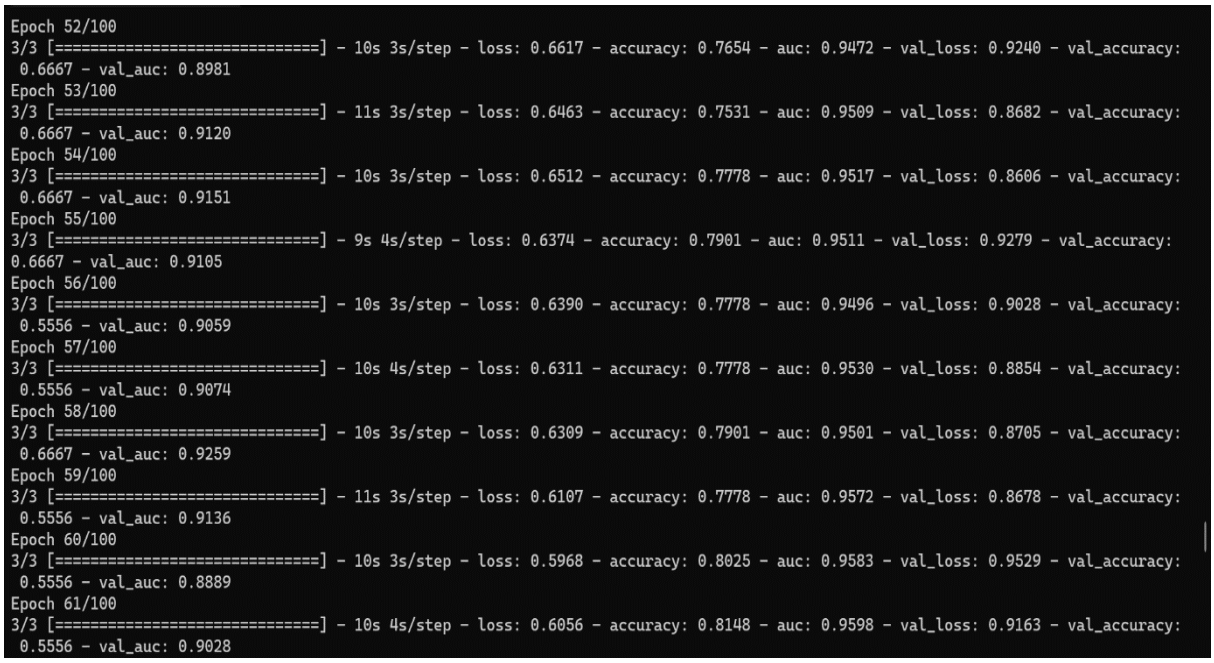

Figure-19 : Result Screen (52-61 epochs)

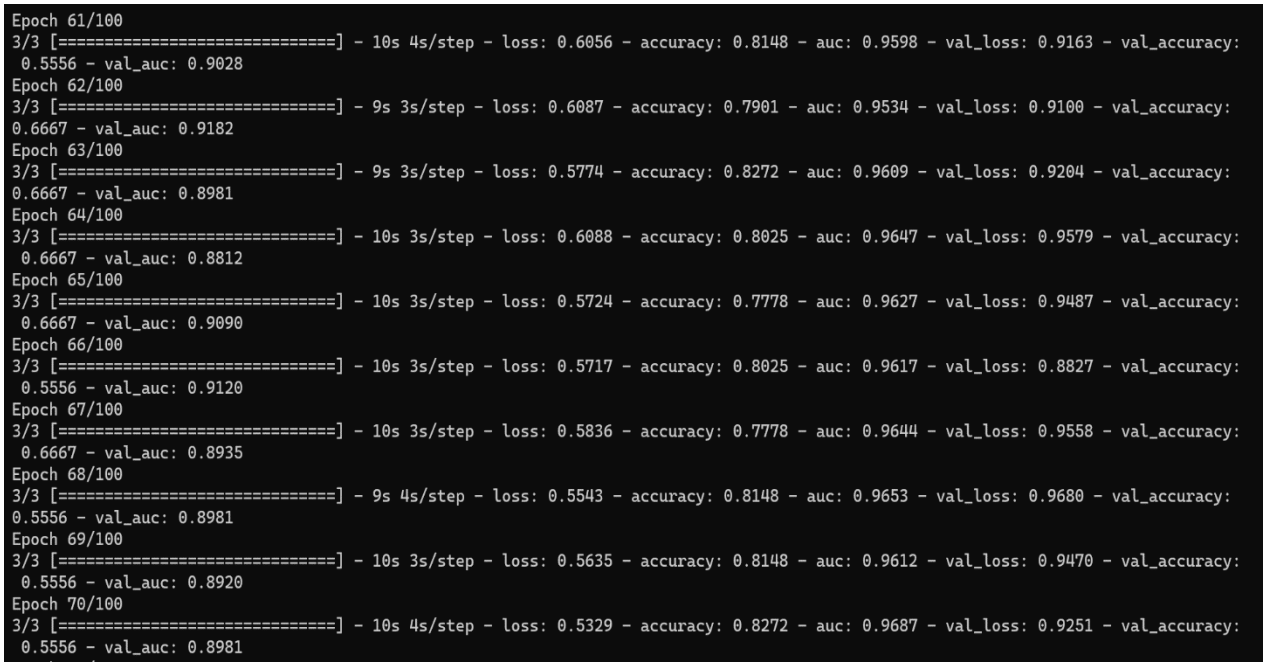

Figure-20 : Result Screen (61-71 epochs)

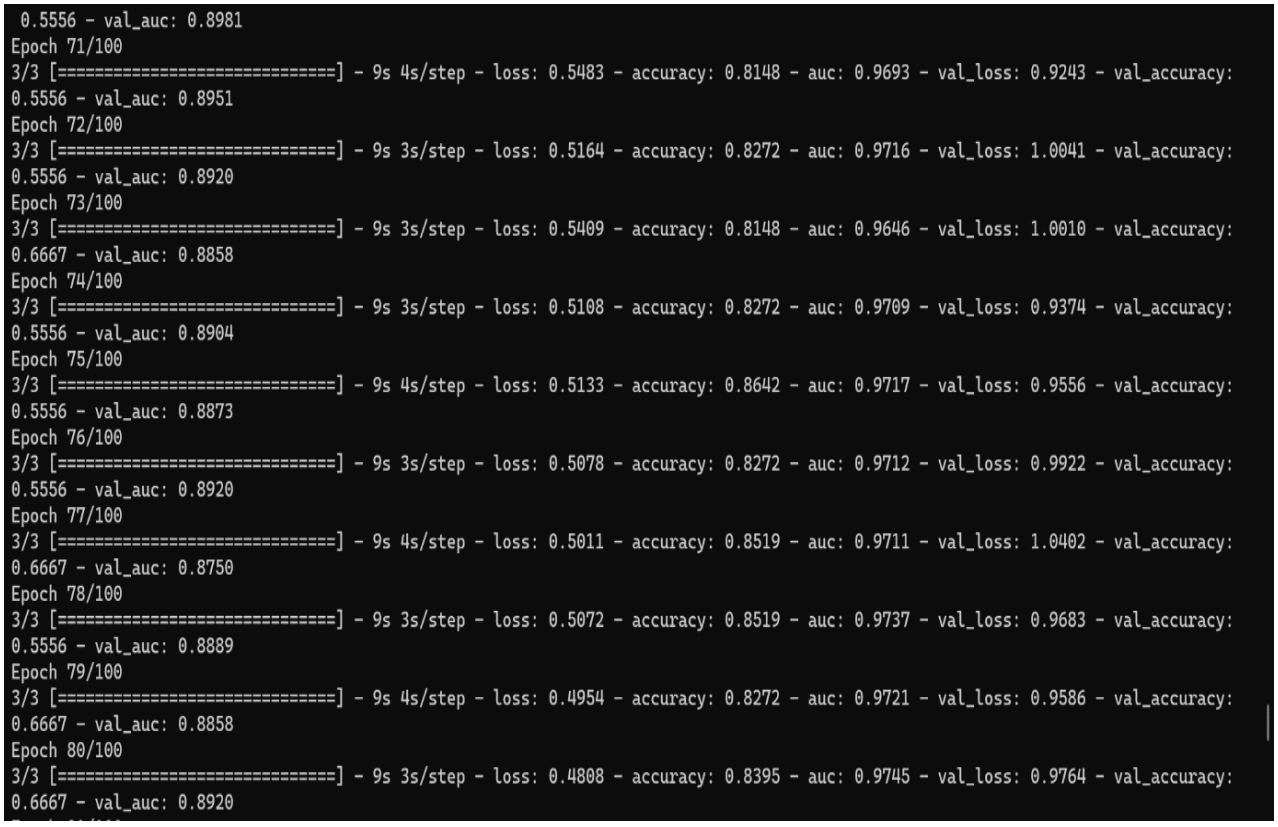

Figure-21 : Result Screen (72-80 epochs)

```
Epoch 81/100
3/3 [==============================] - 9s 4s/step - loss: 0.4715 - accuracy: 0.8765 - auc: 0.9764 - val_loss: 1.0043 - val_accuracy:
0.6667 - val_auc: 0.8765
Epoch 82/100
3/3 [==============================] - 9s 4s/step - loss: 0.4814 - accuracy: 0.8395 - auc: 0.9730 - val_loss: 0.9580 - val_accuracy:
0.5556 - val_auc: 0.8935
Epoch 83/100
3/3 [==============================] - 10s 3s/step - loss: 0.4647 - accuracy: 0.8642 - auc: 0.9765 - val_loss: 1.0258 - val_accuracy:
 0.6667 - val_auc: 0.8827
Epoch 84/100
3/3 [==============================] - 9s 4s/step - loss: 0.4692 - accuracy: 0.8642 - auc: 0.9734 - val_loss: 1.0489 - val_accuracy:
0.6667 - val_auc: 0.8796
Epoch 85/100
3/3 [==============================] - 9s 3s/step - loss: 0.4600 - accuracy: 0.8519 - auc: 0.9774 - val_loss: 1.0354 - val_accuracy:
0.6667 - val_auc: 0.8796
Epoch 86/100
3/3 [==============================] - 9s 3s/step - loss: 0.4446 - accuracy: 0.8889 - auc: 0.9786 - val_loss: 1.0288 - val_accuracy:
0.5556 - val_auc: 0.8812
Epoch 87/100
3/3 [==============================] - 9s 4s/step - loss: 0.4548 - accuracy: 0.8519 - auc: 0.9755 - val_loss: 1.0548 - val_accuracy:
0.6667 - val_auc: 0.8812
Epoch 88/100
3/3 [==============================] - 9s 3s/step - loss: 0.4777 - accuracy: 0.8642 - auc: 0.9773 - val_loss: 1.0269 - val_accuracy:
0.6667 - val_auc: 0.8858
Epoch 89/100
3/3 [==============================] - 10s 3s/step - loss: 0.4917 - accuracy: 0.8395 - auc: 0.9696 - val_loss: 1.0089 - val_accuracy:
 0.5556 - val_auc: 0.8889
Epoch 90/100
3/3 [==============================] - 9s 4s/step - loss: 0.4401 - accuracy: 0.8642 - auc: 0.9797 - val_loss: 1.1553 - val_accuracy:
0.4444 - val_auc: 0.8410
```

Figure-22 : Result Screen (81-90 epochs)

```
Epoch 91/100
3/3 [==============================] - 9s 4s/step - loss: 0.4361 - accuracy: 0.8889 - auc: 0.9819 - val_loss: 1.0905 - val_accuracy:
0.6667 - val_auc: 0.8735
Epoch 92/100
3/3 [==============================] - 9s 3s/step - loss: 0.5002 - accuracy: 0.8148 - auc: 0.9697 - val_loss: 1.0104 - val_accuracy:
0.5556 - val_auc: 0.8889
Epoch 93/100
3/3 [==============================] - 9s 3s/step - loss: 0.5182 - accuracy: 0.8395 - auc: 0.9755 - val_loss: 1.0626 - val_accuracy:
0.5556 - val_auc: 0.8719
Epoch 94/100
3/3 [==============================] - 9s 4s/step - loss: 0.5677 - accuracy: 0.7778 - auc: 0.9648 - val_loss: 1.2006 - val_accuracy:
0.5556 - val_auc: 0.8750
Epoch 95/100
3/3 [==============================] - 9s 3s/step - loss: 0.4290 - accuracy: 0.8519 - auc: 0.9783 - val_loss: 1.3659 - val_accuracy:
0.5556 - val_auc: 0.7593
Epoch 96/100
3/3 [==============================] - 9s 3s/step - loss: 0.6350 - accuracy: 0.7778 - auc: 0.9589 - val_loss: 0.9730 - val_accuracy:
0.6667 - val_auc: 0.8935
Epoch 97/100
3/3 [==============================] - 9s 3s/step - loss: 0.4675 - accuracy: 0.8148 - auc: 0.9735 - val_loss: 1.1428 - val_accuracy:
0.6667 - val_auc: 0.8951
Epoch 98/100
3/3 [==============================] - 9s 3s/step - loss: 0.5168 - accuracy: 0.8148 - auc: 0.9689 - val_loss: 1.1407 - val_accuracy:
0.5556 - val_auc: 0.8611
Epoch 99/100
3/3 [==============================] - 9s 3s/step - loss: 0.4900 - accuracy: 0.8395 - auc: 0.9759 - val_loss: 1.0028 - val_accuracy:
0.4444 - val_auc: 0.8796
Epoch 100/100
3/3 [==============================] - 9s 3s/step - loss: 0.4385 - accuracy: 0.8765 - auc: 0.9744 - val_loss: 1.0702 - val_accuracy:
0.5556 - val_auc: 0.8812
[2 2 2 2 2 1 2 2 2 2 2]
['Moderate', 'No DR', 'Severe', 'No DR', 'Moderate', 'No DR', 'Mild', 'No DR', 'No DR', 'No DR', 'No DR']
```

Figure-23 : Result Screen (91-100 epochs)

The following image displays the sample images of the dataset to be displayed in the resultscreen.

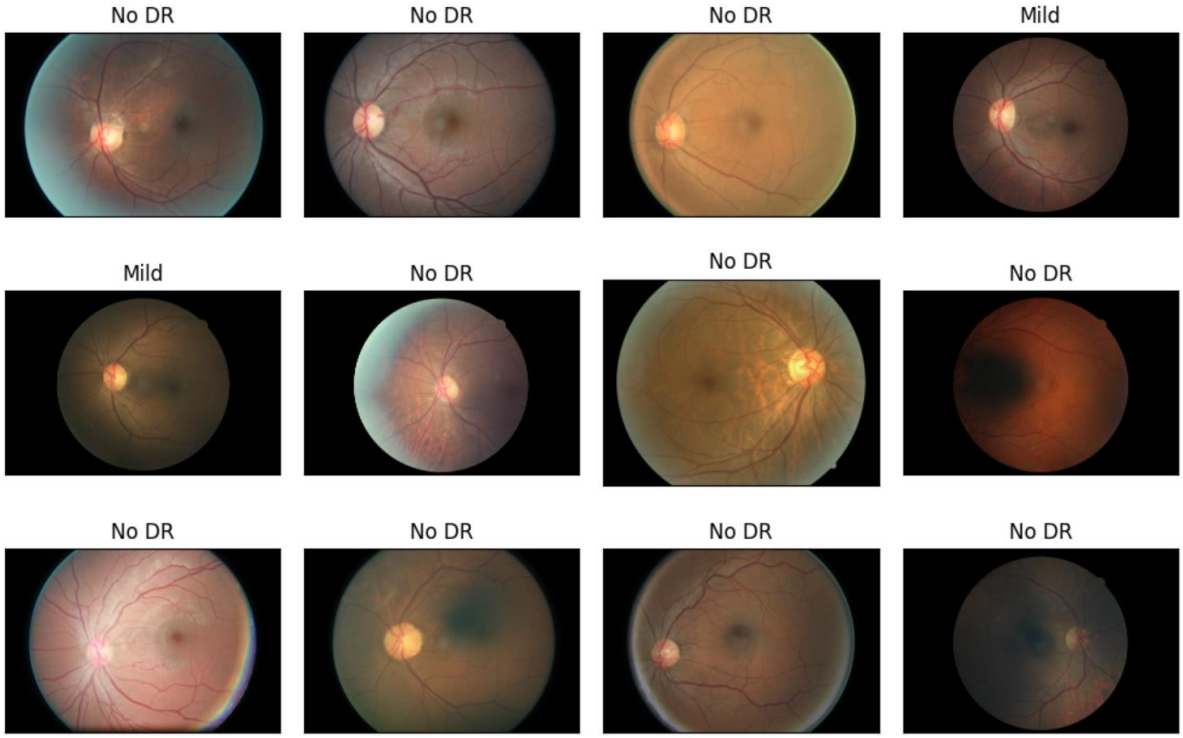

Figure-24 : Result Screen showing sample images

The following screen shows the Accuracy of Training and Test dataset

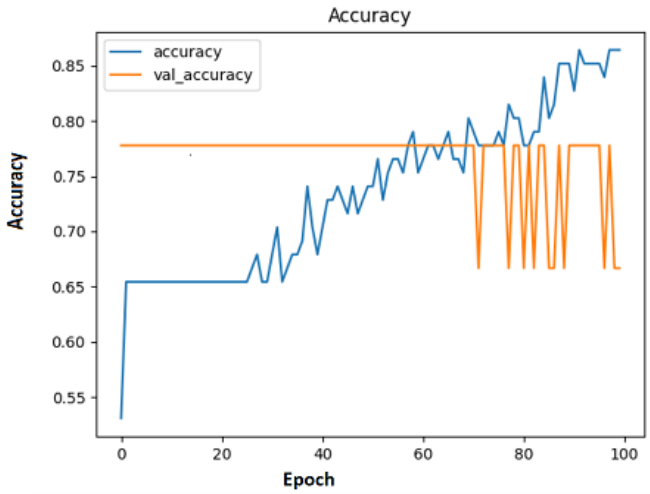

Figure-25 : Accuracy plot

The following screen shows the loss of Training and validation dataset

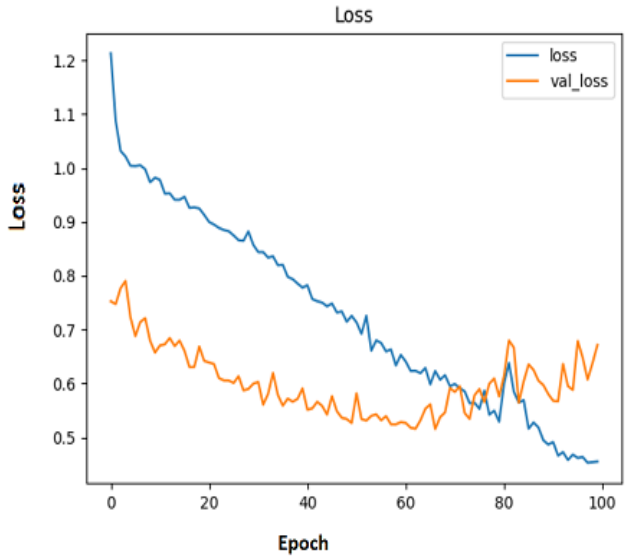

Figure-26 : Loss metric

The following screen shows the AUC metric of Training and validation dataset

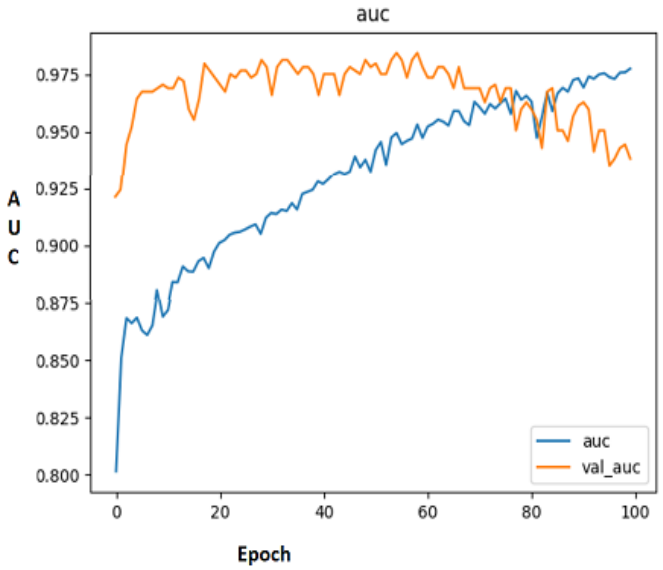

Figure-27 : AUC Metric

The following table represents the training and validation AUC score after every 10 epochs upto 100 epochs.

Epoch	AUC	Validation AUC
1	0.80	0.92
10	0.86	0.97
20	0.89	0.97
30	0.91	0.97
40	0.92	0.96
50	0.93	0.97
60	0.94	0.97
70	0.96	0.96
80	0.96	0.96
90	0.97	0.96
100	0.97	0.94

Table-1 : AUC Scores Obtained

8.CONCLUSION

CONCLUSION

A proposed method for detecting diabetic retinopathy uses the pre-trained ResNet50 deep learning model. Automated screening systems based on deep learning significantly shorten the time needed for diagnosis, save ophthalmologists time and money, and enable quick patient treatment. Automated systems, however, play a significant part in early DR detection. the project that used data augmentation to improve learning capacity. ResNet50 model was employed in the proposed work for validation and training. According to the results of the experiment, the proposed work has a respectable AUC score of 97.77%.

FUTURE ENHANCEMENT

As further enhancement, the work can be extended with more pre-trained model implementation such as AlexNet and VGG16. The work can be also extended with hyper parameter tuning to identify best fit parameter through Grid Search Cross Validation (GSCV) can be implemented in the future work.

9.REFERENCES

[1] D. Sarwinda, T. Siswantining and A. Bustamam, "Classification of Diabetic Retinopathy Stages using Histogram of Oriented Gradients and Shallow Learning," 2018 International Conference on Computer, Control, Informatics and its Applications (IC3INA), 2018, pp. 83-87, doi: 10.1109/IC3INA.2018.8629502.

[2] Colomer, Adrián et al. “Detection of Early Signs of Diabetic Retinopathy Based on Textural and Morphological Information in Fundus Images.” Sensors (Basel, Switzerland) 20 (2020): n. pag.

[3] T. Araújo et al., "Data Augmentation for Improving Proliferative Diabetic Retinopathy Detection in Eye Fundus Images," in IEEE Access, vol. 8, pp. 182462-182474, 2020, doi: 10.1109/ACCESS.2020.3028960.

[4] H. Mustafa, S. F. Ali, M. Bilal and M. S. Hanif, "Multi-Stream Deep Neural Network for Diabetic Retinopathy Severity Classification Under a Boosting Framework," in IEEE Access, vol. 10, pp. 113172-113183, 2022, doi: 10.1109/ACCESS.2022.3217216.

[5] A. Herliana, T. Arifin, S. Susanti and A. B. Hikmah, "Feature Selection of Diabetic Retinopathy Disease Using Particle Swarm Optimization and Neural Network," 2018 6th International Conference on Cyber and IT Service Management (CITSM), 2018, pp. 1-4, doi: 10.1109/CITSM.2018.8674295.

[6] Wu L, Fernandez-Loaiza P, Sauma J, Hernandez-Bogantes E, Masis M. Classification of diabetic retinopathy and diabetic macular edema. World J Diabetes. 2013;4(6):290-294. doi:10.4239/wjd.v4.i6.290

[7] Zheng, Yingfeng et al. “The worldwide epidemic of diabetic retinopathy.” Indian journal of ophthalmology vol. 60,5 (2012): 428-31. doi:10.4103/0301-4738.100542

[8] Teo ZL, Tham YC, Yu M, Chee ML, Rim TH, Cheung N, Bikbov MM, Wang YX, Tang Y, Lu Y, Wong IY, Ting DSW, Tan GSW, Jonas JB, Sabanayagam C, Wong TY, Cheng CY. Global Prevalence of Diabetic Retinopathy and Projection of Burden through 2045: Systematic Review and Meta-analysis. Ophthalmology. 2021 Nov;128(11):1580-1591. doi: 10.1016/j.ophtha.2021.04.027. Epub 2021 May 1. PMID: 33940045.

[9] Q. Li and X. Wang, "Image Classification Based on SIFT and SVM," 2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS), 2018, pp. 762-765, doi: 10.1109/ICIS.2018.8466432.

[10] Morales, Y & Nuñez, R & Suarez, J & Torres Moreno, Cesar. (2017). Digital tool for detecting diabetic retinopathy in retinography image using gabor transform. Journal of Physics: Conference Series. 792. 012083. 10.1088/1742-6596/792/1/012083.

[11] Tsiknakis, Nikos & Theodoropoulos, Dimitris & Manikis, Georgios & Ktistakis, Emmanouil & Boutsora, Ourania & Berto, Alexa & Scarpa, Fabio & Scarpa, Alberto & Fotiadis, Dimitrios & Marias, Kostas. (2021). Deep Learning for Diabetic Retinopathy Detection and Classification Based on Fundus Images: A Review. Computers in Biology and Medicine. 135. 104599. 10.1016/j.compbiomed.2021.104599.

Acceptance Letter

5NANO2023 notification for paper 2541

5NANO2023 <5nano2023@easychair.org> 20 February 2023 at 16:47
To: Lakshmi Gayathri Popuri <gayathri20021@gmail.com>

Congratulations,
Greetings from IEEE 5NANO 2023!
We are pleased to inform that your paper has been accepted for the publication in the 2023 IEEE International Conference on Nanoelectronics, Nanophotonics, Nanomaterials, Nanobioscience and Nanotechnology(5NANO 2023)
Kindly send your final camera ready paper and registration on or before 25.02.2023
The reviewers' comments are included with this notification. Please address all the review comments while preparing your final camera ready manuscript.
We would also like to encourage you to submit your final paper early to avoid potential uploading difficulties that may arise due to the heavy uploading activities near the deadline.
All accepted and presented papers will be submitted for inclusion in IEEE Xplore.
Based on the scope and authors concern the extended version of high quality research articles will be recommended for the publication in SCI & SCOPUS indexed Journals without any additional fee.

At least one author of an accepted paper is required to register at the full registration rate. If an author has got more than one accepted papers, each paper has also to be registered.
It may be ensured that the Remitter's (Participant/Author) name, Easychair Paper-ID and the Purpose of remittance (Registration Fees) is clearly mentioned by the Remitter in the Funds Transfer Application. Indian Nationals or Students residing in overseas have to pay foreign authors' registration rate only.

Student Registration: Students (UG/PG Students and only Full time Research/PhD Scholars) have to submit their scanned copy of the ID card along with the Registration form to avail the deduction. When the students are coming for the presentation, they have to produce the original Student ID card.
IEEE Members shall produce evidence of valid current membership at the venue of the Conference. If they fail to do so, they would be required to pay the difference in fees before being allowed to participate at the conference.

Once the payment is made, you are required to scan the proof of Bank Draft or NFET/ RTGS Receipt or Wire Transfer Receipt and upload the details online using the same username and password you had created earlier or through email: 5nano2k23@gmail.com
If you have any questions or clarifications on registration, please email to: 5nano2k23@gmail.com
As it may take some time to get Indian Visa, we request the foreign participants to initiate the visa application as early as possible to attend the conference and visit Kerala, one of the most beautiful places in the world as declared by the National Geographic Traveler in "50 greatest destinations list". We would be glad to support you with any required documents/invitations to get Indian Visa.
Thank you for submitting your paper to IEEE 5NANO 2023 - We look forward to seeing you in Kerala! God's Own Country.
Regards
IEEE 5NANO 2023 Conference Organizing Chair
Registration Fee Details:
Kindly check the link

https://www.5nano2023.com/Registration.php#main

Conference Organizing Chair
IEEE 5NANO 2023
VISAT Engineering College
Elanji, Ernakulam
Kerala, India, Pin - 686 665
Tel: +91 9447691397,
+91 9486881397
E-mail: tdsubash2007@gmail.com, deanresearch@visat.ac.in
5nano2k23@gmail.com

SUBMISSION: 2541
TITLE: A RESNET-50 MODEL FOR THE DETECTION OF DIABETIC RETINOPATHY