DAYANANDA SAGAR UNIVERSITY

KUDLU GATE, BANGALORE – 560068

Bachelor of Technology
in
COMPUTER SCIENCE AND ENGINEERING

Major Project Phase-II Report

Automated Detection of Diabetic Retinopathy using Deep Learning

By

CHAITANYA J M REDDY – ENG18CS0064
GREESHMA C SHEKAR – ENG18CS0108
H V RAKSHITHA – ENG18CS0110
HARSHITHA C S – ENG18CS0114
KEERTHANA R – ENG18CS0131

Under the supervision of
Prof. GEETHA A
Assistant Professor Department Of Computer Science & Engineering

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING,
SCHOOL OF ENGINEERING
DAYANANDA SAGAR UNIVERSITY,

(2021-2022)

DAYANANDA SAGAR UNIVERSITY

School of Engineering
Department of Computer Science & Engineering
Kudlu Gate, Bangalore – 560068
Karnataka, India

CERTIFICATE

This is to certify that the Phase-II project work titled **" AUTOMATED DETECTION OF DIABETIC RETINOPATHY USING DEEP LEARNING "** is carried out by **Chaitanya J M Reddy (ENG18CS0064) , Greeshma C Shekar (EN18CS0108) , H V Rakshitha (ENG18CS0110) , Harshitha C S (ENG18CS0114) , Keerthana R (ENG18CS0131),** bonafide students of Bachelor of Technology in Computer Science and Engineering at the School of Engineering, Dayananda Sagar University, Bangalore in partial fulfillment for the award of degree in Bachelor of Technology in Computer Science and Engineering, during the year **2021-2022**.

Prof Geetha.A Assistant Professor Dept of CS&E, School of Engineering Dayananda Sagar University	**Dr Girisha G S** Chairman CSE School of Engineering Dayananda Sagar University	**Dr. A Srinivas** Dean School of Engineering Dayananda Sagar University
Date:	Date:	Date:

Name of the Examiner **Signature of Examiner**

1.

2.

DECLARATION

We, **Chaitanya J M Reddy (ENG18CS0064), Greeshma C Shekar (ENG18CS0108), H V Rakshitha (ENG18CS0110), Harshitha C S(ENG18CS0114), Keerthana R (ENG18CS0131),** are student's of the eighth semester B.Tech in **Computer Science and Engineering**, at School of Engineering, **Dayananda Sagar University**, hereby declare that the phase-II project titled **"Automated Detection of Diabetic Retinopathy using Deep Learning"** has been carried out by us and submitted in partial fulfillment for the award of degree in **Bachelor of Technology in Computer Science and Engineering** during the academic year **2021-2022**.

Student	**Signature**
Chaitanya J M Reddy	
ENG18CS0064	
Greeshma C Shekar	
ENG18CS0108	
H V Rakshitha	
ENG18CS0110	
Harshitha C S	
ENG18CS0114	
Keerthana R	
ENG18CS0131	

Place : Bangalore

Date :

ACKNOWLEDGEMENT

It is a great pleasure for us to acknowledge the assistance and support of many individuals who have been responsible for the successful completion of this project work.

First, we take this opportunity to express our sincere gratitude to School of Engineering & Technology, Dayananda Sagar University for providing us with a great opportunity to pursue our Bachelor's degree in this institution.

We would like to thank **Dr. A Srinivas. Dean**, **School of Engineering & Technology**, **Dayananda Sagar University** for his constant encouragement and expert advice.It is a matter of immense pleasure to express our sincere thanks to **Dr. Girisha G S, Department Chairman**, **Computer Science, and Engineering**, **Dayananda Sagar University,** for providing the right academic guidance that made our task possible.

We would like to thank our guide **Prof Geetha.A Assistant Professor**, **Department of Computer Science and Engineering**, **Dayananda Sagar University**, for sparing her valuable time to extend help in every step of our project work, which paved the way for smooth progress and the fruitful culmination of the project.

We would like to thank our Project Coordinators **Dr. Meenakshi Malhotra and Dr. Bharanidharan N,** and all the staff members of Computer Science and Engineering for their support.

We are also grateful to our family and friends who provided us with every requirement throughout the course. We would like to thank one and all who directly or indirectly helped us in the Project work.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

MySQL	Structured Query Language
DL	Deep Learning
GUI	Graphical User Interface
DR	Diabetic retinopathy
CNN	Convolutional neural network
RESNET	Residual neural network
VGG-16	Visual geometry group of CNN
NPDR	Non-Proliferative Diabetic retinopathy
PDR	Proliferative Diabetic retinopathy

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Diabetic retinopathy and diabetic macular edema result from chronic damage to the neurovascular structures of the retina. The pathophysiology of retinal damage remains uncertain but includes metabolic and neuroinflammatory insults. These mechanisms are addressed by intensive metabolic control of the systemic disease and by the use of ocular anti-inflammatory agents, including vascular endothelial growth factor inhibitors and corticosteroids. Improved understanding of the ocular and systemic mechanisms that underlie diabetic retinopathy will lead to improved means to diagnose and treat retinopathy and better maintain vision.

Diabetic retinopathy and diabetic macular edema result from chronic damage to the neuro vascular structures of the retina. Our project uses a deep learning classification technique using CNN pre-trained model resnet152 to classify severity levels of DR ranging from 0 which means no presence of diabetic retinopathy to 4 which means presence of proliferative diabetic retinopathy. To provide access to ocular care to the low-income population with diabetes and to offer early assessment and referral for timely treatment. To establish a health-care network for the referral of patients with diabetic retinopathy for screening, evaluation and ocular diagnosis, and treatment if required. To strengthen the technical capacity of medical teams, providing adequate training to improve the efficiency and quality of diabetic retinopathy services.

Key Words: Diabetic Retinopathy, Convolutional Neural Network, VGG-16 and ResNet152v2

CHAPTER 1

INTRODUCTION

CHAPTER 1 INTRODUCTION

1.1 PURPOSE

Diabetic retinopathy is a leading cause of blindness among working-age adults. It is the fastest growing cause of preventable blindness. All people with diabetes are at risk. They need to be screened once a year. In many parts of the world there's a shortage of eye doctors. As a result, in India about 45% of people suffer some form of vision loss before the disease is detected. In this project, we demonstrate the use of convolutional neural networks (CNNs) on color fundus images for the recognition of diabetic retinopathy. Automated techniques for diabetic retinopathy diagnoses are essential to solving these problems. While deep learning for binary classification in general has achieved high validation accuracies, multi- stage classification results are less impressive, particularly for early-stage disease.

A convolutional neural network (CNN) convolves an input image with a defined weight matrix to extract specific image features without losing spatial arrangement information. We then seek to train multi-class models that enhance sensitivities for the mild or early- stage classes, including various methods of data preprocessing and data augmentation to both improve test accuracy as well as increase our effective dataset sample size. Finally, we address the issue of insufficient sample size using a deep layered CNN with transfer learning on discriminant color space for the recognition task

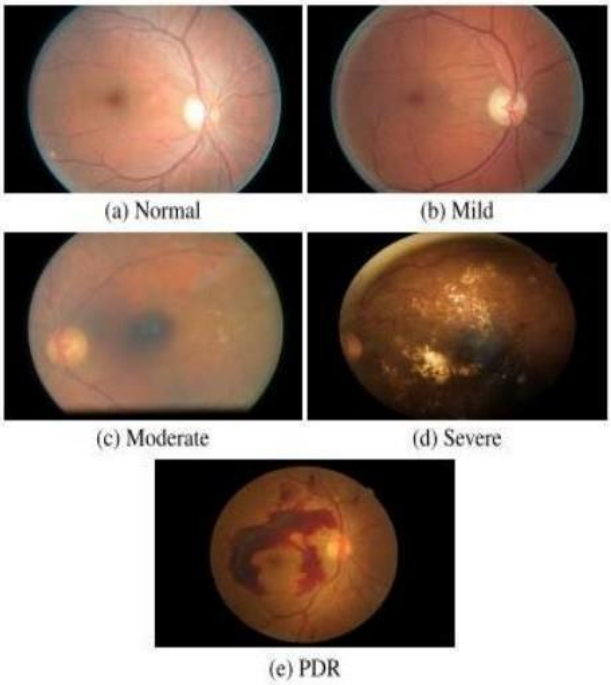


Figure 1: Stages of Diabetic Retinopathy

1.2 SCOPE OF PROJECT

(i) To provide access to ocular care to the low-income population with diabetes and to offer early

assessment and referral for timely treatment.

(ii) To establish a health-care network for the referral of patients with diabetic retinopathy for screening, evaluation and ocular diagnosis, and treatment if required.

(iii) To strengthen the technical capacity of medical teams, providing adequate training to improve the efficiency and quality of diabetic retinopathy services.

Diabetes was once thought of as a disease of the affluent but it has now reached epidemic proportion in both developed and developing countries. Currently, at least 366 million people worldwide have diabetes, and this number is likely to increase as a result of an aging global population

Globally, the number of people with DR will grow from 126.6 million in 2010 to 191.0 million by 2030, and we estimate that the number with vision-threatening diabetic retinopathy (VTDR) will increase from 37.3 million to 56.3 million, if prompt action is not taken.

CHAPTER 2

PROBLEM DEFINITION

CHAPTER 2 PROBLEM DEFINITION

Diabetic Retinopathy is a disease with an increasing prevalence and the main cause of blindness among the working-age population. The risk of severe vision loss can be significantly reduced by timely diagnosis andtreatment. Systematic screening for diabetic retinopathy has been identified as a cost- effective way to save health services resources. Automatic retinal image analysis isemerging as an important screening tool for early DR detection, which can reduce the workload associated with manual grading as well as save diagnosis costs and time. Many research efforts in the last years have been devoted to developing automated tools to help in the detection and evaluation of DR lesions and are interested in automating this prediction using deep learning models.To classify severity levels of DR, we are proposing Deep Learning classification technique using CNN pretrained model. Diabetic Retinopathy is a disease with an increasing prevalence and the main cause of blindness and an early diagnosis and treatment is necessary.To classify severity levels of DR, we are proposing Deep Learning classification technique using CNN, Resnet and VGG-16 models.

CHAPTER 3

LITERATURE REVIEW

CHAPTER 3 LITERATURE REVIEW

1) **Matias Iglicki, MD, PHD, Dinah Zur, MD, and Anat Loewenstein, MD, MHA, are the authors of "Detection of Diabetic Retinopathy Using Deep Learning Analysis" published in retina today 2021 feature.**

With advances in digital image processing and communications, the authors believe telemedicine can become a viable screening tool for patients at risk for developing diabetic retinopathy (DR).

A deep-learning telemedicine platform designed by the authors achieved statistically significant sensitivities, specificities, and positive predictive values for both referrable and vision-threatening DR. A follow-up study is planned to further assess the system's ability to automatically detect hard exudates and hemorrhages compared with traditional examination techniques.

2) **S. Yu, D. Xiao and Y. Kanagasingam, "Exudate detection for diabetic retinopathy with convolutional neural networks," 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2017, pp. 1744-1747, doi: 10.1109/EMBC.2017.8037180.**

Exudate detection is an essential task for computer-aid diagnosis of diabetic retinopathy (DR), so as to monitor the progress of DR. In this paper, deep convolutional neural network (CNN) is adopted to achieve pixel-wise exudate identification. The CNN model is first trained with expert labeled exudates image patches and then saved as off-line classifier.

Typically, the detection of exudates can be broadly divided into three steps: getting exudate candidates, extracting features and machine learning.

Since optic disc and exudates are both bright objects and share similar color in color fundus image, as a result, before effective exudates detection, the optic disc shall be detected and

masked out. Next is the Removal of Retinal Vessels Generally, the removal of retinal vessels are performed by vessel segmentation and then inpainting the vessel based on the segmentation result. After the retinal vessel and small dark lesions are removed, In order to achieve pixel-level accuracy potential exudate candidate points are first extracted with morphological ultimate opening algorithm..A pixel-wise accuracy of 91.92%, sensitivity of 88.85% and specificity of 96% is achievedwith the proposed CNN architecture on the test database.

3) **Yashal Shakti Kanungo, Bhargav Srinivasan, Dr. Savita Choudhary , "Detecting Diabetic Retinopathy using Deep Learning" , 2017 2nd IEEE International Conference On Recent Trends in Electronics Information & Communication Technology (RTEICT), May 19-20, 2017, India**

The diagnosis can be automated by using Convolutional Neural Networks. Convolution Neural Networks have abilities to identify patterns directly from image at pixel level and hence requires minimal preprocessing. It provides advantage over others by recognizing patterns under extreme variability In this journal, a deep learning algorithm was used for automated detection of diabetic retinopathy in retinal fundus images.

The deep convolutional neural network had high sensitivity and specificity deep learning proved to be an exciting potential which was not fully explored and it motivated us to work on this problem statement using this approach. The architecture basically acts as multiple convolution filter inputs that are processed on the same input. It also does pooling at the same time. All the results are then concatenated. This allows the model to take advantage of multi-level feature extraction from each The architecture focuses on not only improving the accuracy and statistical performance of the model but also on the most efficient way to do so considering the training time and memory footprint

4) **Kavakiotis, I.; Tsave, O.; Salifoglou, A.; Maglaveras, N.; Vlahavas, I.; Chouvarda, I. Machine learning and data mining methods in diabetes research. Comput. Struct. Biotechnol. J. 2017, 15, 104–116.**

People with vision-threatening diabetic retinopathy are likely to experience enhanced social and emotional strain. Critically, those with both vision-threatening diabetic retinopathy and psychosocial problems may have significantly reduced levels of functioning compared with psychologically healthy counterparts. This can cause inadequate compliance, increased strain on family functioning, worse diabetes control, increased progression of diabetic retinopathy and, consequently, further psychosocial stress resulting in a number of concerning implications for disease management, clinical outcomes and healthcare costs. However, the emotional and social health consequences of diabetic retinopathy have not yet been systematically explored.

Adverse emotional responses include fear, anxiety, vulnerability, guilt, loss of confidence, anger, stress and self-perception issues. However, the research to date is largely qualitative in nature, with most quantitative studies being small, cross-sectional and somewhat outdated. Similarly, the outcome measures used in many studies to date are suboptimal in terms of content and validity.

5) **S. Qummar et al., "A Deep Learning Ensemble Approach for Diabetic 10.1109/ACCESS.2019.2947484.**

A ensemble based framework has been proposed to detect and classify different stages of Dr in colour fundus images. Stacking is a model used to combine information from multiple predictive models to generate a new model which they implemented in the project. The trends illustrated by them shows that if the learning rate reduces from 0.01 to 1e-05 the recall and accuracy improves, but the specificity is affected due to misclassification of the positive class.

They have focused to classify all the stages of DR, especially the early stages, which is the major shortcoming of existing models. They used the largest publicly available

dataset of fundus images (Kaggle dataset) to train and evaluate their model. The results have showed that the proposed ensemble model performs better than other state-of-the-art methods and is also able to detect all the stages of DR.

6) **W. Chen, B. Yang, J. Li and J. Wang, "An Approach to Detecting Diabetic Retinopathy Based on Integrated Shallow Convolutional Neural Networks," in IEEE Access, vol. 8, pp. 178552-178562, 2020, doi: 10.1109/ACCESS.2020.3027794.**

In this paper, they have proposed a shallow CNNs based integrated learning model, the model proposed is composed of L base learners, each of which acts as a shallow CNN to learn the image features under a specific vision-related receptive field , that only represent the details of 3 out of these base learners, such as their convolutional layers, pooling layers and full connected layers. The output of all learners will be integrated according to a policy proposed to enhance the sensing to multi-scale features and improving the classification.

They have owed to the feature sensing under various vision-related receptive fields by different base learners and the repeatable dataset sampling, so that it can do image classification well when there are not enough high-quality labelled samples. As a result of their experiments, the performance integration model showed an advantage in accuracy compared with other integration models like that based on Mean and Voting. Moreover, their proposed approach also seems to be performed well on small datasets when considering of both classification effect and efficiency compared with other approaches.

CHAPTER 4

PROJECT DESCRIPTION

CHAPTER 4 PROJECT DESCRIPTION

4.1 PROPOSED DESIGN

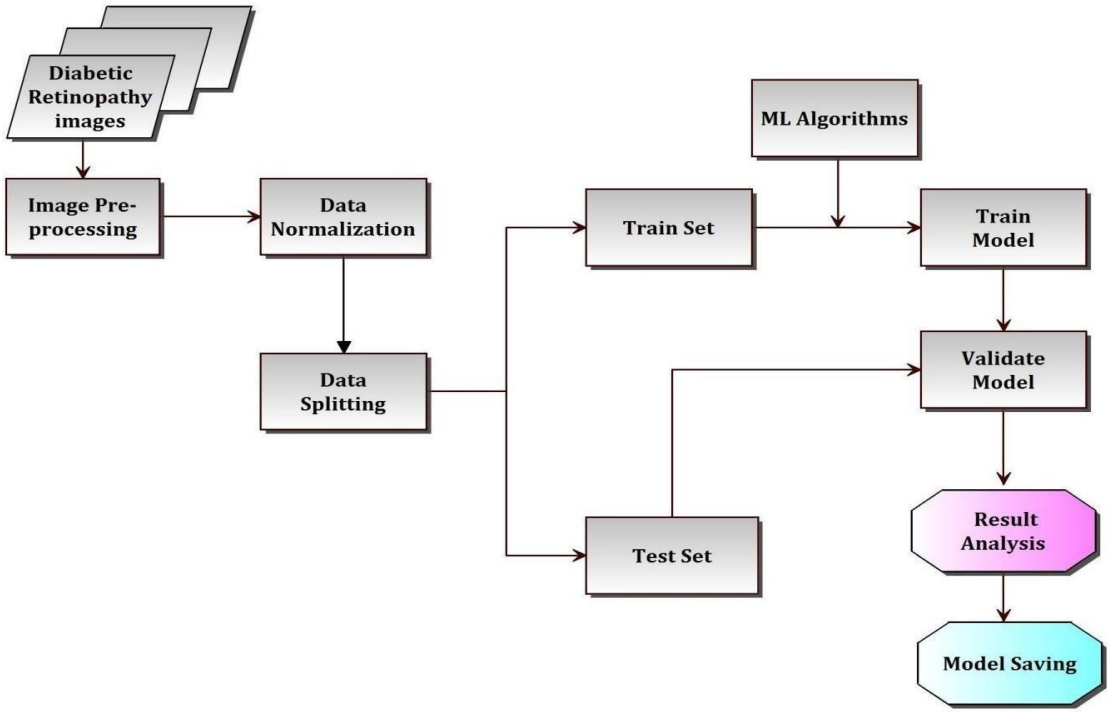


Figure 4.1 Proposed Design

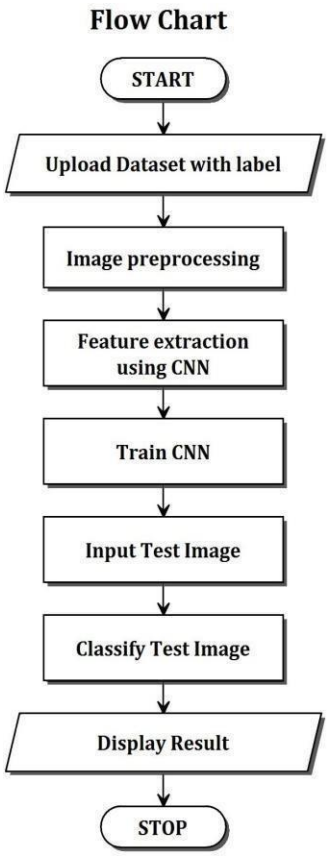


Figure 4.2 Flow Chart Diagram

4.2 ASSUMPTIONS AND DEPENDENCIES

- The application requires a stable network connection.
- The user must have a google account so as to use run the application
- The application will always be used on operating systems and browsers that have enough performance.

CHAPTER 5

REQUIREMENTS

CHAPTER 5 REQUIREMENTS

5.1 FUNCTIONAL REQUIREMENTS

The Functional requirements for the proposed system are:

- Developing GUI using python GUI Toolkit.
- Gathering the data which includes the images of the diabetic retinopathy.
- Implementing image classification detection of diabetic retinopathy using Deep learning algorithm.
- Implementing backend using one Deep learning framework
- Using MySQL database connectivity.
- Integrating the GUI with the backend.

5.2 NON-FUNCTIONAL REQUIREMENT

5.2.1 Reliability

The structure must be reliable and strong in giving the functionalities. The movements must be made unmistakable by the structure when a customer has revealed a couple of enhancements. The progressions made by the Programmer must be Project pioneer and in addition the Test designer.

5.2.2 Maintainability

The system watching and upkeep should be fundamental and focus in its approach. There should not be an excess of occupations running on diverse machines such that it gets hardto screen whether the employments are running without lapses.

5.2.3 Performance

The framework will be utilized by numerous representatives all the while. Since the systemwill be encouraged on a single web server with a lone database server outside of anyone's ability to see, execution transforms into a significant concern. The structure should not capitulate when various customers would use everything the while. It should allow brisk accessibility to each and every piece of its customers. For instance, if two test specialists are all the while attempting to report the vicinity of a bug, then there ought not to be any irregularity at the same time.

5.2.4 Portability

The framework should to be effectively versatile to another framework. This is obliged when the web server, which s facilitating the framework gets adhered because of a few issues, which requires the framework to be taken to another framework.

5.2.5 Scalability

The framework should be sufficiently adaptable to include new functionalities at a later stage. There should be a run of the mill channel, which can oblige the new functionalities.

5.2.6 Flexibility

Flexibility is the capacity of a framework to adjust to changing situations and circumstances, and to adapt to changes to business approaches and rules. An adaptable framework is one that is anything but difficult to reconfigure or adjust because of diverse client and framework prerequisites. The deliberate division of concerns between the trough and motor parts helps adaptability as just a little bit of the framework is influenced when strategies or principles change.

5.3 SOFTWARE REQUIREMENTS

- My SQL
- PyCharm IDE
- Anaconda Navigator
- Jupyter Notebook
- Python flask (web server)
- Google Chrome (web client)

5.4 HARDWARE REQUIREMENTS

- 8 GB RAM and above recommended.
- Processor: Intel core i5 or above
- Ram: 8GB (OR) above
- Storage: Minimum 5 GB disk space to download and install.
- Operating system: Windows 7 or newer, 64-bit macOS 10.13+, or Linux,
- including Ubuntu, RedHat, CentOS 6+, and others.
- System architecture: Windows- 64-bit x86, 32-bit x86; MacOS- 64-bit x86;
- Linux- 64-bit x86, 64-bit Power8/Power9.

CHAPTER 6

METHODOLOGY

CHAPTER 6 METHODOLOGY

In this project, CNN, ResNet152V2 and VGG-16 Approach for the implementation.

Step 1: Dataset and Labels

The data collected by the Asia-Pacific TeleOphthalmology Society (APTOS) available on the Kaggle platform. The data set consists of 3,662 color fundus images.

The dataset of such images for the training and testing of our model.Each image in the dataset has been assigned an integral value on the scale of 0 to 4 according to the severity of the disease by a professionally trained clinician as shown in Table-1.

Table 6: Severity levels of Diabetic Retinopathy

Class	Name	No of images
0	Normal	1805
1	Mild Non-Proliferative DR	370
2	Moderate Non-Proliferative DR	999
3	Severe Non-Proliferative DR	193
4	Proliferative DR	295

0 – Normal: The person is not suffering from Diabetic Retinopathy.

1 – Mild Non-Proliferative DR (Mild_NPDR): Within the Retina's minute blood vessels, small areas of balloon like inflammations.

2 – Moderate Non-Proliferative DR (Moderate_NPDR): The blood vessels that sustain the retina are blocked at this stage. Within the retina, there might also be hemorrhages.

3 – Severe Non-Proliferative DR (Severe_NPDR): More blood vessels are blocked in this particular stage, denying several areas of retina of blood supply. The number of hemorrhages in the retina also increases drastically.

4 – Proliferative DR (PDR): New and abnormal blood vessels developed on the surface of retina. These new blood vessels are delicate and have the tendency to bleed, causing vision threatening hemorrhages to fill the eye. They can also turn into connective tissue which will contract over time, causing the retina to detach and cause blindness.

Step 2 – Preprocessing the Dataset

The Resizing of images for pre-processing the data before providing it to the model first, keeping in view the aspect ratio to 256 × 256. It helps us to avoid features lossfrom images. The dataset is distributed into training, testing sets with a ratio of 80% and 20% respectively.

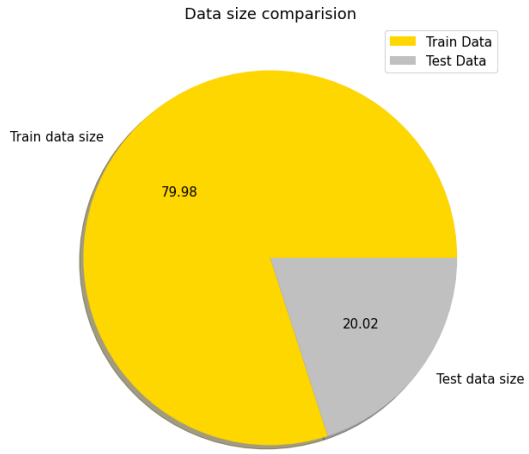


Figure 6.1 Data size comparison

Step 3 – Splitting the Data into Train and Test Set

In order to induce a more balanced date, the dataset split the dataset into 80:20 for training and validation purpose. Totally for training purpose we have considered2,929 color fundus images and 733 color fundus images for validation.

```
train
0 1444
1 296
2 799
3 154
4 236
train ----- 2929
val
0 361
1 74
2 200
3 39
4 59
val ----- 733
```

Figure 6.2 Preview Of Dataset

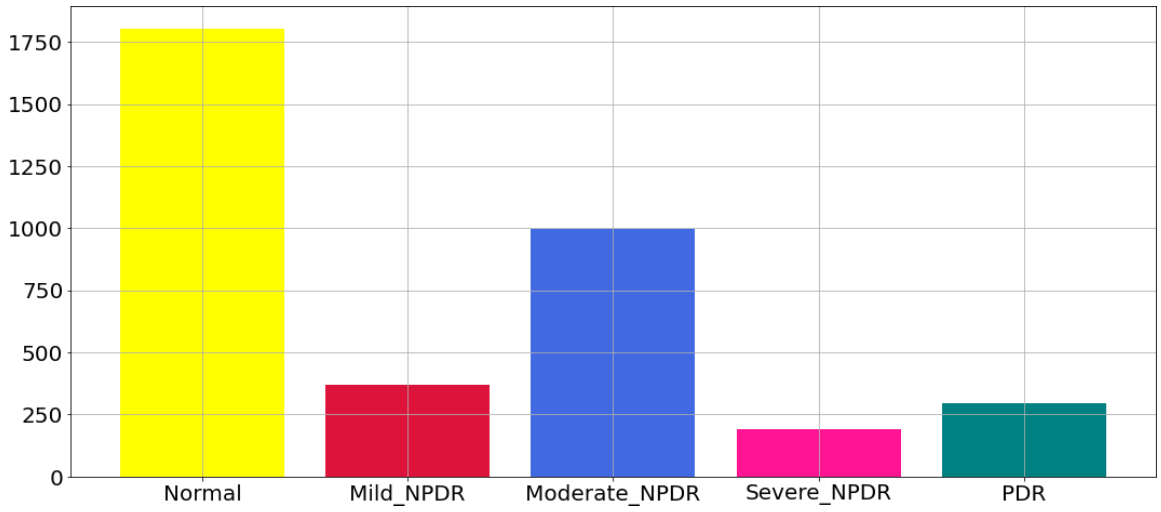


Figure 6.3 Train Data Size

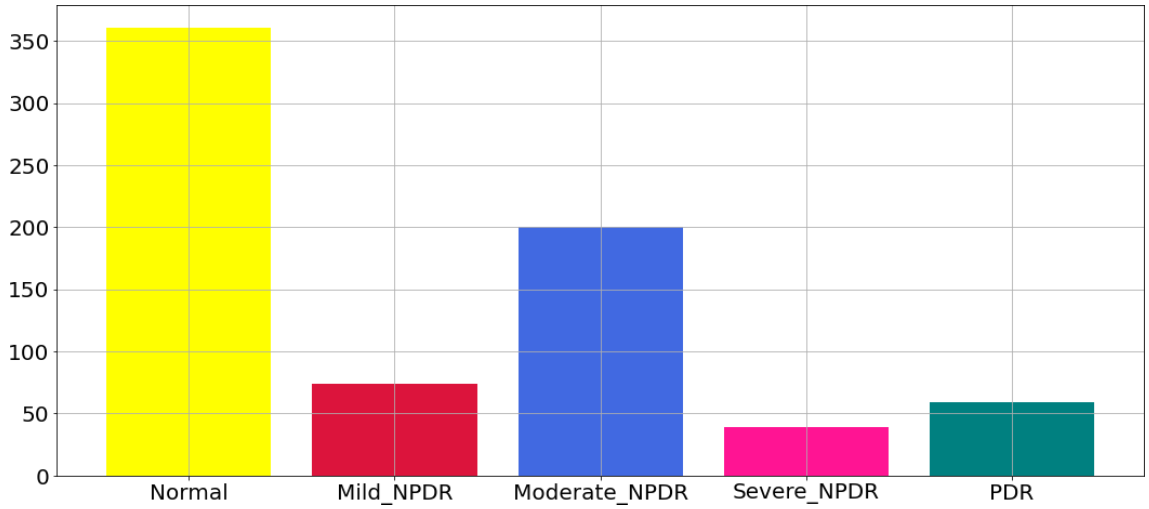


Figure 6.4 Test Data Size

Step 4 – Model Building

Following the data split, the train data is fed to various models in order to train them. The models used are:

CNN: Convolutional neural network

RESNET: Residual neural network

VGG-16: Visual geometry group of CNN

Model Building Of CNN

```
In [24]: model=Sequential()

model.add(Conv2D(64,(3,3),input_shape=x_train.shape[1:],padding="same",activation="relu"))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Conv2D(64,(3,3),padding="same",activation="relu"))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Conv2D(128,(3,3),padding="same",activation="relu"))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Conv2D(128,(3,3),padding="same",activation="relu"))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.2))
model.add(Flatten())
model.add(Dense(1024,activation='relu'))
model.add(Dense(5,activation='softmax'))
```

Model Building Of VGG-16

The pretrained model is taken from ImageNet and a fully connected layer was added.

In [19]:
```
import tensorflow as tf
```

In [20]:
```
vgg_model=tf.keras.applications.VGG16(include_top=False,
                                      weights="imagenet",
                                      input_tensor=None,
                                      input_shape=(256,256,3),
                                      pooling="max",
                                      classes=1000)
vgg_model.trainable = False
```

In [21]:
```
inputs = vgg_model.input

m = tf.keras.layers.Dense(1024, activation='relu')(vgg_model.output)

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

vgg_model = tf.keras.Model(inputs=inputs, outputs=outputs)
```

In [22]:
```
vgg_model.compile(optimizer='adam',
                  loss='categorical_crossentropy',
                  metrics=['accuracy'])
```

Model Building Of RESENT152V2

In [19]:
```
import tensorflow as tf
```

In [20]:
```
resnet_model=tf.keras.applications.ResNet152V2(include_top=False,
                                               weights="imagenet",
                                               input_tensor=None,
                                               input_shape=(256,256,3),
                                               pooling="max",
                                               classes=1000)
resnet_model.trainable = False
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/resnet/resnet152v2_weights_tf_dim_ordering_tf_kernels_notop.h5
234553344/234545216 [==============================] - 21s 0us/step
234561536/234545216 [==============================] - 21s 0us/step
```

In [23]:
```
inputs = resnet_model.input

m = tf.keras.layers.Dense(1024, activation='relu')(resnet_model.output)

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

resnet_model = tf.keras.Model(inputs=inputs, outputs=outputs)
```

In [24]:
```
resnet_model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])
```

CHAPTER 7

EXPERIMENTATION

CHAPTER 7 EXPERIMENTATION

7.1 Data splitting

The creation of different samples for training and testing helps us evaluate model performance. Hence the split of our modelling dataset into training and testing samples is performed using the train_test_split() function of the scikit-learn library. The train and test dataset splitted previously is again split into x_train,x_test,y_test and y_train,that is visualized below in figure.

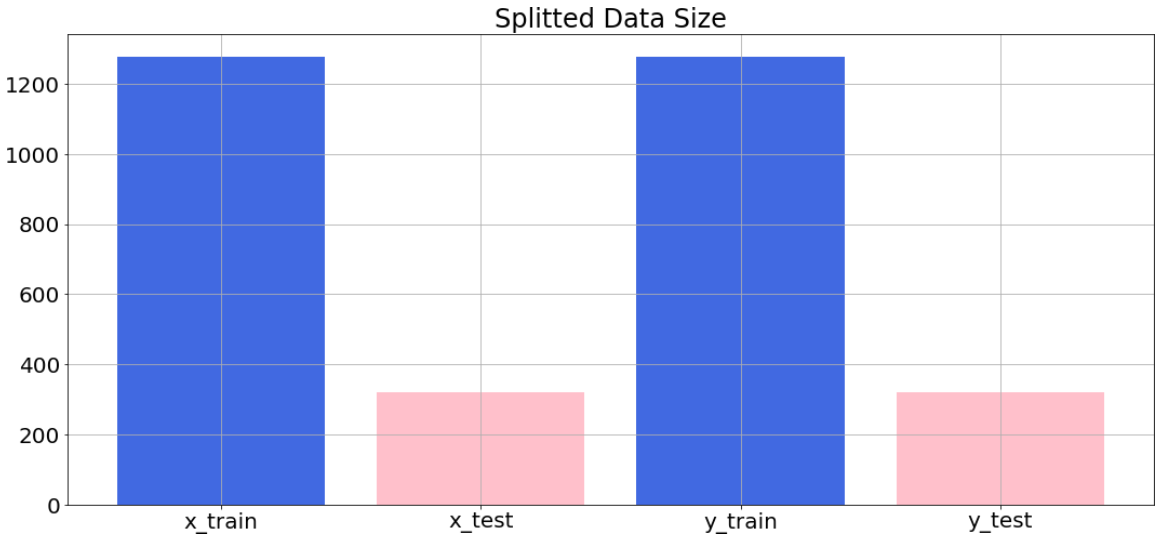


Figure7.1 Splitted Data Size

7.2 Accuracy Comparison Of 3 Models

Various models as previously discussed were used and the accuracy and performance of those are compared. The various models used their train accuracies are provided below inTable 7.1.

Table 7.1 Accuracy Comparison Of 3 Models

Model	Train Accuracy	Test Accuracy
RESENT152V2	**98.51**	**94**
VGG16	**68**	**69**
CNN	**92.41**	**89**

7.3 Model Summary

```
In [23]: vgg_model.summary()
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 256, 256, 3)]     0
 block1_conv1 (Conv2D)       (None, 256, 256, 64)      1792
 block1_conv2 (Conv2D)       (None, 256, 256, 64)      36928
 block1_pool (MaxPooling2D)  (None, 128, 128, 64)      0
 block2_conv1 (Conv2D)       (None, 128, 128, 128)     73856
 block2_conv2 (Conv2D)       (None, 128, 128, 128)     147584
 block2_pool (MaxPooling2D)  (None, 64, 64, 128)       0
 block3_conv1 (Conv2D)       (None, 64, 64, 256)       295168
 block3_conv2 (Conv2D)       (None, 64, 64, 256)       590080
 block3_conv3 (Conv2D)       (None, 64, 64, 256)       590080
 block3_pool (MaxPooling2D)  (None, 32, 32, 256)       0
 block4_conv1 (Conv2D)       (None, 32, 32, 512)       1180160
 block4_conv2 (Conv2D)       (None, 32, 32, 512)       2359808
 block4_conv3 (Conv2D)       (None, 32, 32, 512)       2359808
 block4_pool (MaxPooling2D)  (None, 16, 16, 512)       0
 block5_conv1 (Conv2D)       (None, 16, 16, 512)       2359808
 block5_conv2 (Conv2D)       (None, 16, 16, 512)       2359808
 block5_conv3 (Conv2D)       (None, 16, 16, 512)       2359808
 block5_pool (MaxPooling2D)  (None, 8, 8, 512)         0
 global_max_pooling2d (Globa  (None, 512)              0
 lMaxPooling2D)
 dense (Dense)               (None, 1024)              525312
 dense_1 (Dense)             (None, 5)                 5125
=================================================================
Total params: 15,245,125
Trainable params: 530,437
Non-trainable params: 14,714,688
```

Figure 7.2 Model Summary of Vgg16

```
In [26]: model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 256, 256, 64)      1792

 max_pooling2d (MaxPooling2D  (None, 128, 128, 64)     0
 )

 conv2d_1 (Conv2D)           (None, 128, 128, 64)      36928

 max_pooling2d_1 (MaxPooling  (None, 64, 64, 64)       0
 2D)

 conv2d_2 (Conv2D)           (None, 64, 64, 128)       73856

 max_pooling2d_2 (MaxPooling  (None, 32, 32, 128)      0
 2D)

 conv2d_3 (Conv2D)           (None, 32, 32, 128)       147584

 max_pooling2d_3 (MaxPooling  (None, 16, 16, 128)      0
 2D)

 dropout (Dropout)           (None, 16, 16, 128)       0

 flatten (Flatten)           (None, 32768)             0

 dense (Dense)               (None, 1024)              33555456

 dense_1 (Dense)             (None, 5)                 5125

=================================================================
Total params: 33,820,741
Trainable params: 33,820,741
Non-trainable params: 0
_________________________________________________________________
```

Figure 7.3 Model Summary of CNN

7.4 MODEL TRAINING

7.4.1 RESENT152V2

```
In [26]: history=resnet_model.fit(x=x_train,y=y_train,batch_size=64,epochs=20)
```

```
Epoch 1/20
20/20 [==============================] - 470s 23s/step - loss: 17.8685 - accuracy: 0.3020
Epoch 2/20
20/20 [==============================] - 454s 23s/step - loss: 1.9330 - accuracy: 0.5078
Epoch 3/20
20/20 [==============================] - 456s 23s/step - loss: 1.1251 - accuracy: 0.5790
Epoch 4/20
20/20 [==============================] - 456s 23s/step - loss: 0.8334 - accuracy: 0.6581
Epoch 5/20
20/20 [==============================] - 461s 23s/step - loss: 0.6829 - accuracy: 0.7488
Epoch 6/20
20/20 [==============================] - 454s 23s/step - loss: 0.5897 - accuracy: 0.7879
Epoch 7/20
20/20 [==============================] - 457s 23s/step - loss: 0.5129 - accuracy: 0.8216
Epoch 8/20
20/20 [==============================] - 456s 23s/step - loss: 0.4611 - accuracy: 0.8584
Epoch 9/20
20/20 [==============================] - 453s 23s/step - loss: 0.3948 - accuracy: 0.8889
Epoch 10/20
20/20 [==============================] - 459s 23s/step - loss: 0.3292 - accuracy: 0.9030
Epoch 11/20
20/20 [==============================] - 455s 23s/step - loss: 0.3016 - accuracy: 0.9171
Epoch 12/20
20/20 [==============================] - 456s 23s/step - loss: 0.2398 - accuracy: 0.9570
Epoch 13/20
20/20 [==============================] - 455s 23s/step - loss: 0.1891 - accuracy: 0.9640
Epoch 14/20
20/20 [==============================] - 452s 23s/step - loss: 0.1550 - accuracy: 0.9726
Epoch 15/20
20/20 [==============================] - 452s 23s/step - loss: 0.1547 - accuracy: 0.9679
Epoch 16/20
20/20 [==============================] - 453s 23s/step - loss: 0.1244 - accuracy: 0.9797
Epoch 17/20
20/20 [==============================] - 454s 23s/step - loss: 0.1030 - accuracy: 0.9851
```

7.4.2 VGG16

```
In [24]: history=vgg_model.fit(x=x_train,y=y_train,batch_size=64,epochs=20)
```

```
Epoch 1/20
20/20 [==============================] - 521s 26s/step - loss: 1.6168 - accuracy: 0.3779
Epoch 2/20
20/20 [==============================] - 515s 26s/step - loss: 1.2670 - accuracy: 0.4804
Epoch 3/20
20/20 [==============================] - 517s 26s/step - loss: 1.1910 - accuracy: 0.5000
Epoch 4/20
20/20 [==============================] - 516s 26s/step - loss: 1.1302 - accuracy: 0.5509
Epoch 5/20
20/20 [==============================] - 514s 26s/step - loss: 1.1089 - accuracy: 0.5391
Epoch 6/20
20/20 [==============================] - 515s 26s/step - loss: 1.0530 - accuracy: 0.5806
Epoch 7/20
20/20 [==============================] - 514s 26s/step - loss: 1.0371 - accuracy: 0.5970
Epoch 8/20
20/20 [==============================] - 522s 26s/step - loss: 1.0554 - accuracy: 0.5767
Epoch 9/20
20/20 [==============================] - 520s 26s/step - loss: 1.0171 - accuracy: 0.6009
Epoch 10/20
20/20 [==============================] - 519s 26s/step - loss: 0.9504 - accuracy: 0.6244
Epoch 11/20
20/20 [==============================] - 520s 26s/step - loss: 0.9403 - accuracy: 0.6315
Epoch 12/20
20/20 [==============================] - 517s 26s/step - loss: 0.9302 - accuracy: 0.6369
Epoch 13/20
20/20 [==============================] - 520s 26s/step - loss: 0.8955 - accuracy: 0.6651
Epoch 14/20
20/20 [==============================] - 518s 26s/step - loss: 0.9196 - accuracy: 0.6455
Epoch 15/20
20/20 [==============================] - 519s 26s/step - loss: 0.9019 - accuracy: 0.6440
Epoch 16/20
20/20 [==============================] - 518s 26s/step - loss: 0.8495 - accuracy: 0.6620
Epoch 17/20
20/20 [==============================] - 519s 26s/step - loss: 0.8419 - accuracy: 0.6800
Epoch 18/20
```

7.4.3 CNN

```
In [27]: history=model.fit(x=x_train,y=y_train,batch_size=64,epochs=20)
```

```
Epoch 1/20
20/20 [==============================] - 188s 9s/step - loss: 1.5173 - accuracy: 0.3568
Epoch 2/20
20/20 [==============================] - 186s 9s/step - loss: 1.2298 - accuracy: 0.5156
Epoch 3/20
20/20 [==============================] - 186s 9s/step - loss: 1.1959 - accuracy: 0.4977
Epoch 4/20
20/20 [==============================] - 185s 9s/step - loss: 1.1513 - accuracy: 0.5462
Epoch 5/20
20/20 [==============================] - 186s 9s/step - loss: 1.1253 - accuracy: 0.5516
Epoch 6/20
20/20 [==============================] - 185s 9s/step - loss: 1.1048 - accuracy: 0.5775
Epoch 7/20
20/20 [==============================] - 186s 9s/step - loss: 1.0253 - accuracy: 0.6080
Epoch 8/20
20/20 [==============================] - 185s 9s/step - loss: 0.9633 - accuracy: 0.6291
Epoch 9/20
20/20 [==============================] - 186s 9s/step - loss: 0.8782 - accuracy: 0.6596
Epoch 10/20
20/20 [==============================] - 185s 9s/step - loss: 0.8475 - accuracy: 0.6737
Epoch 11/20
20/20 [==============================] - 185s 9s/step - loss: 0.7316 - accuracy: 0.7230
Epoch 12/20
20/20 [==============================] - 185s 9s/step - loss: 0.6220 - accuracy: 0.7692
Epoch 13/20
20/20 [==============================] - 187s 9s/step - loss: 0.5244 - accuracy: 0.7997
Epoch 14/20
20/20 [==============================] - 188s 9s/step - loss: 0.4127 - accuracy: 0.8521
Epoch 15/20
20/20 [==============================] - 185s 9s/step - loss: 0.3126 - accuracy: 0.8944
Epoch 16/20
20/20 [==============================] - 186s 9s/step - loss: 0.2602 - accuracy: 0.9108
Epoch 17/20
20/20 [==============================] - 186s 9s/step - loss: 0.2417 - accuracy: 0.9241
Epoch 18/20
```

CHAPTER 8

TESTING AND RESULTS

CHAPTER 8 TESTING AND RESULTS

8.1 Diabetic Retinopathy Prediction By The Model On The Test Data

A prediction with values from test dataset is predicted. Figure shows the results of predictions.

```
[30]: y_pred = model.predict(x_test, batch_size=64, verbose=1)
```

```
[31]: print(y_pred)
```

```
[30]: y_pred = model.predict(x_test, batch_size=64, verbose=1)
5/5 [==============================] - 13s 2s/step
[31]: print(y_pred)
[[9.9977499e-01 3.5483263e-05 5.8709422e-05 1.0513023e-04 2.5618276e-05]
 [9.9666411e-01 3.0599318e-03 2.7571130e-04 2.4822369e-09 1.2495775e-07]
 [9.9984038e-01 5.9145546e-05 2.2849290e-05 7.7296456e-05 2.8444208e-07]
 ...
 [1.6630594e-05 5.0106242e-02 8.9847755e-01 1.5854253e-04 5.1241037e-02]
 [1.0599570e-07 1.2925740e-07 3.9197144e-04 5.1559436e-01 4.8401347e-01]
 [4.0846739e-02 8.9072919e-01 2.5254684e-03 6.7528628e-04 6.5223217e-02]]
```

The models true values are also printed to compare the true negatives ,true positives,false negatives and false positives to have a holistic view of the model.

```
[35]: y_true = []
      for i in range(0, len(y_test)):
          y_true.append(int(np.argmax(y_test[i])))
      len(y_true)

[35]: 320

[36]: print(y_true)

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4]
```

8.2 Diabetic Retinopathy Classification Report

8.2.1 Resent152v2

If the accuracy is high and the loss is low, then the model makes small errors on just someof the data, which would be the ideal case. The following plots in Fig 8.1 show the loss atand accuracy of the RESENT152V2 model on training sets

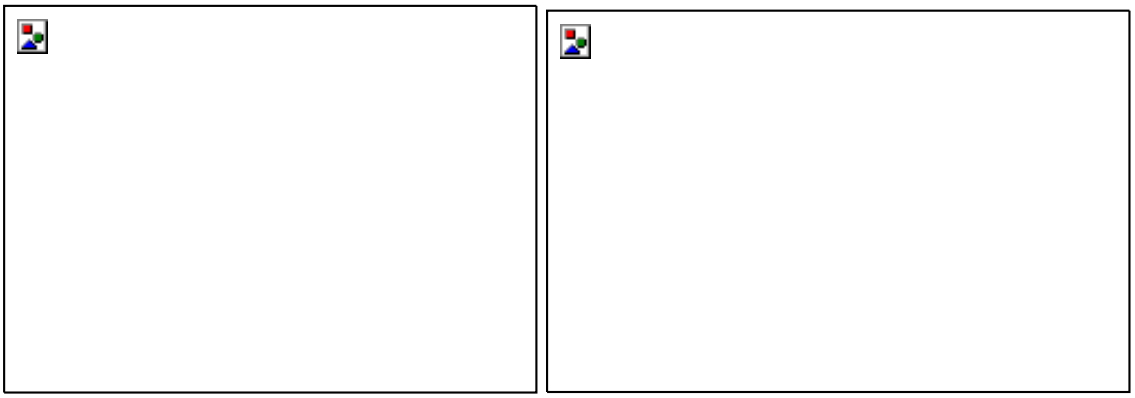


Figure 8.1 The learning curve for Resent152v2

A classification report is a performance evaluation metric in machine learning. It is used toshow the precision, recall, and F1 Score of your trained classification model. Fig 8.1 showsthe trained model's classification report.

Table 8.1. Classification Report Of RESNET152V2

Levels	Precision	Recall	F1-Score	Support
Normal	0.97	1.00	0.99	74
Mild_NPDR	0.91	0.92	0.91	74
Moderate_NPDR	0.99	0.93	0.96	74
Severe_NPDR	0.88	0.97	0.93	39
PDR	0.91	0.86	0.89	59
Accuracy			0.94	320
Macro avg	0.93	0.94	0.93	320
Weighted avg	0.94	0.94	0.94	320

8.2.2 VGG16

If the accuracy is high and the loss is low, then the model makes small errors on just some of the data, which would be the ideal case. The following plots in Fig 8.2 show the loss atand accuracy of the VGG16 model on training sets

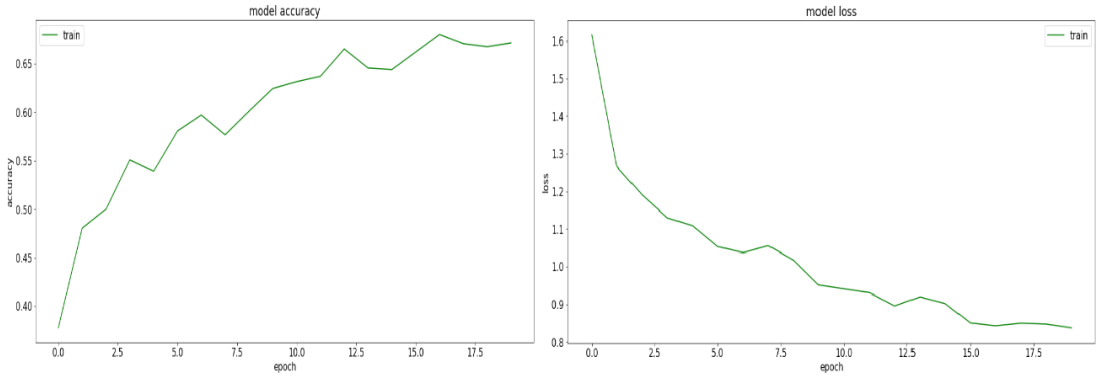


Figure 8.2 The learning curve forVGG16

A classification report is a performance evaluation metric in machine learning. It is used toshow the precision, recall, and F1 Score of your trained classification model. Fig 8.2 showsthe trained model's classification report.

Table 8.2 Classification Report Of VGG16

Levels	**Precision**	**Recall**	**F1-Score**	**Support**
Normal	0.93	0.92	0.93	74
Mild_NPDR	0.59	0.88	0.71	74
Moderat_NPDR	0.60	0.59	0.60	74
Severe_NPDR	0.64	0.54	0.58	39
PDR	0.74	0.39	0.51	59
Accuracy			0.69	320
Macro avg	0.70	0.566	0.66	320
Weighted avg	0.71	0.69	0.68	320

8.2.3 CNN

If the accuracy is high and the loss is low, then the model makes small errors on just some of the data, which would be the ideal case. The following plots in Fig 8.3 show the loss at and accuracy of the CNN model on training sets

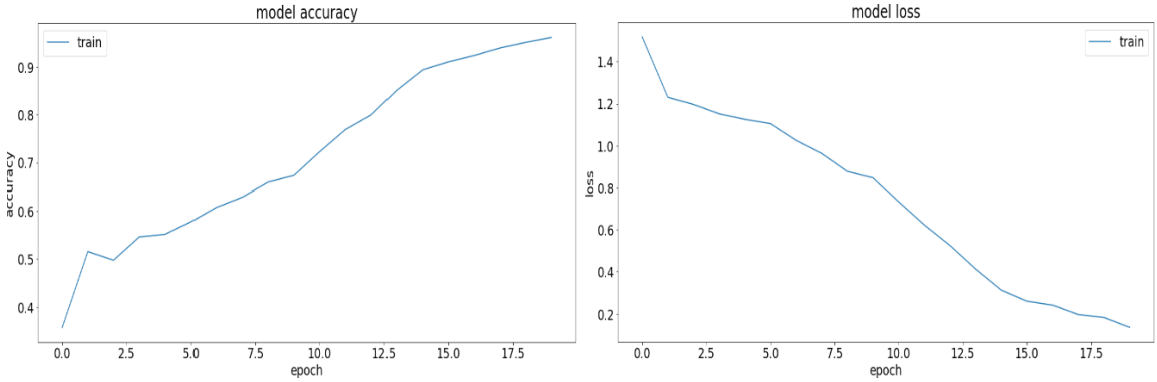


Figure 8.3 The learning curve for CNN

A classification report is a performance evaluation metric in machine learning. It is used toshow the precision, recall, and F1 Score of your trained classification model. Fig 8.3 showsthe trained model's classification report.

Table 8.3 Classification Report Of CNN

Levels	Precision	Recall	F1-Score	Support
Normal	0.94	0.99	0.96	74
Mild_NPDR	0.88	0.88	0.88	74
Moderate_NPDR	0.86	0.86	0.86	74
Severe_NPDR	0.88	0.92	0.90	39
PDR	0.91	0.81	0.86	59
Accuracy			0.89	320
Macro avg	0.89	0.89	0.89	320
Weighted avg	0.89	0.89	0.89	320

CONFUSION MATRIX

A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model. The matrix compares the actual target values with those predicted by the machine learning model. This gives us a holistic view of how well our classification model is performing and what kinds of errors it is making. Fig 8.4 shows the confusion matrix of the CNN, RESENT152V1, VGG16 Model.

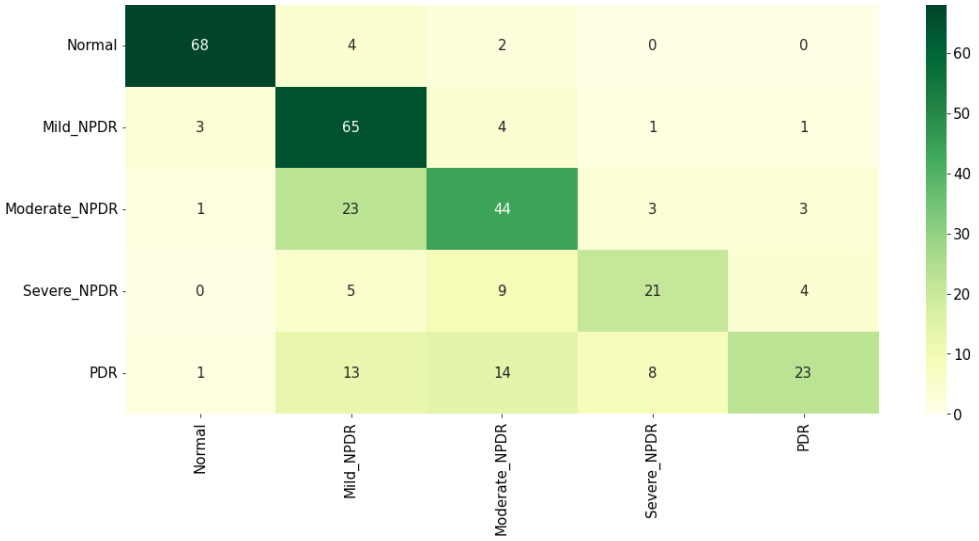


Figure 8.4 Confusion Matrix Of The VGG16

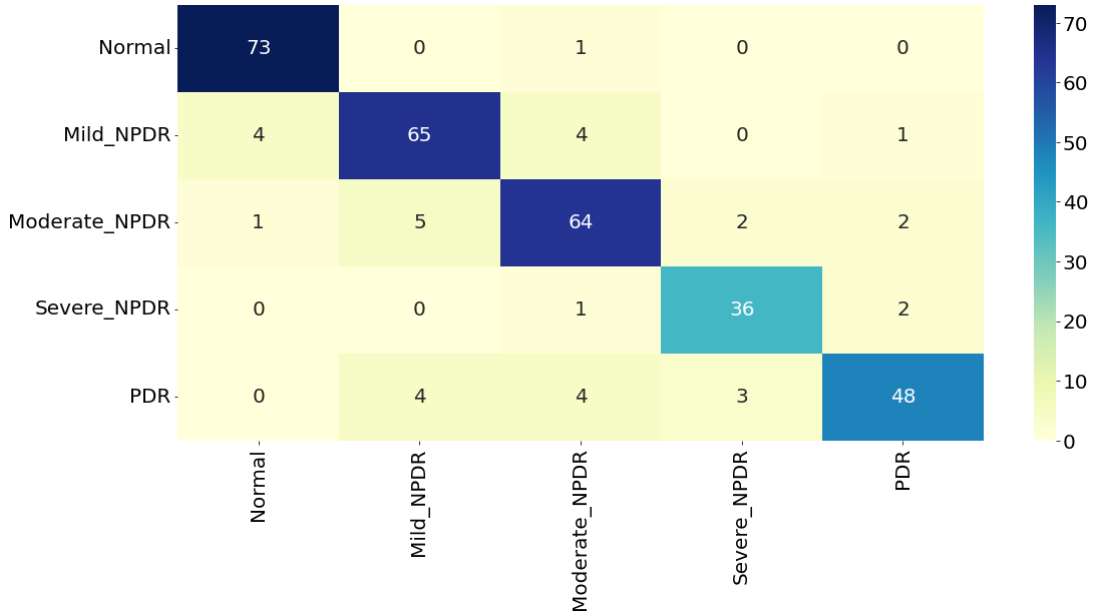


Fig. 8.5 Confusion Matrix of CNN

Resnet152v2

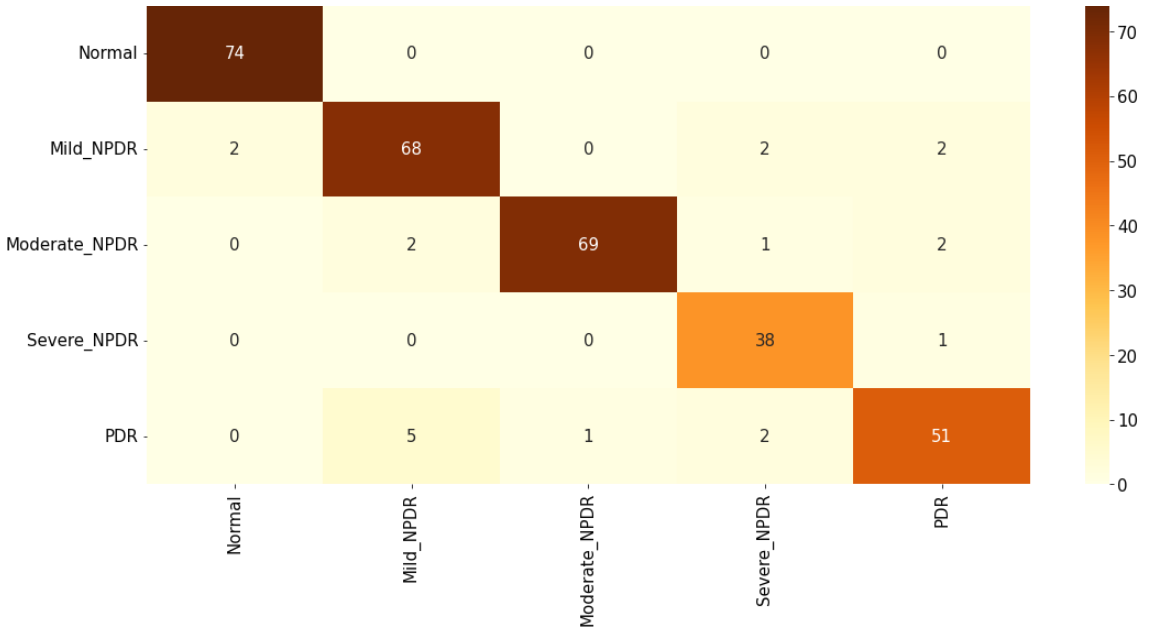


Fig. 8.6 Confusion Matrix of RESNET152V2

CHAPTER 9

OUTPUTS

CHAPTER 9 OUTPUTS

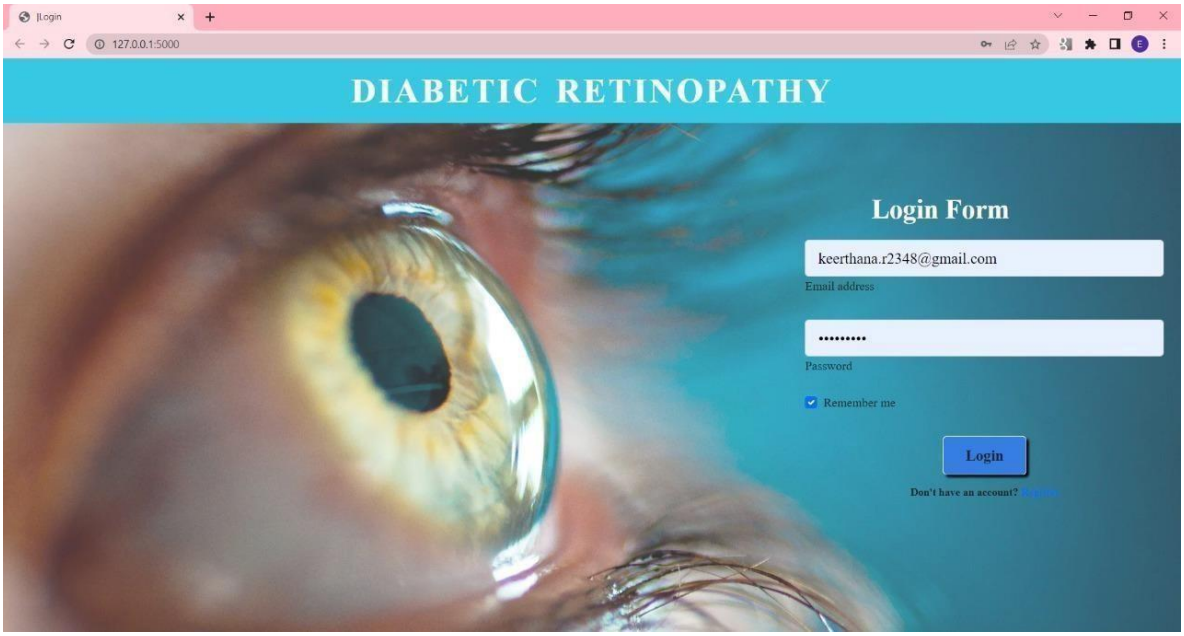


FIGURE9.1 USER LOGIN PAGE

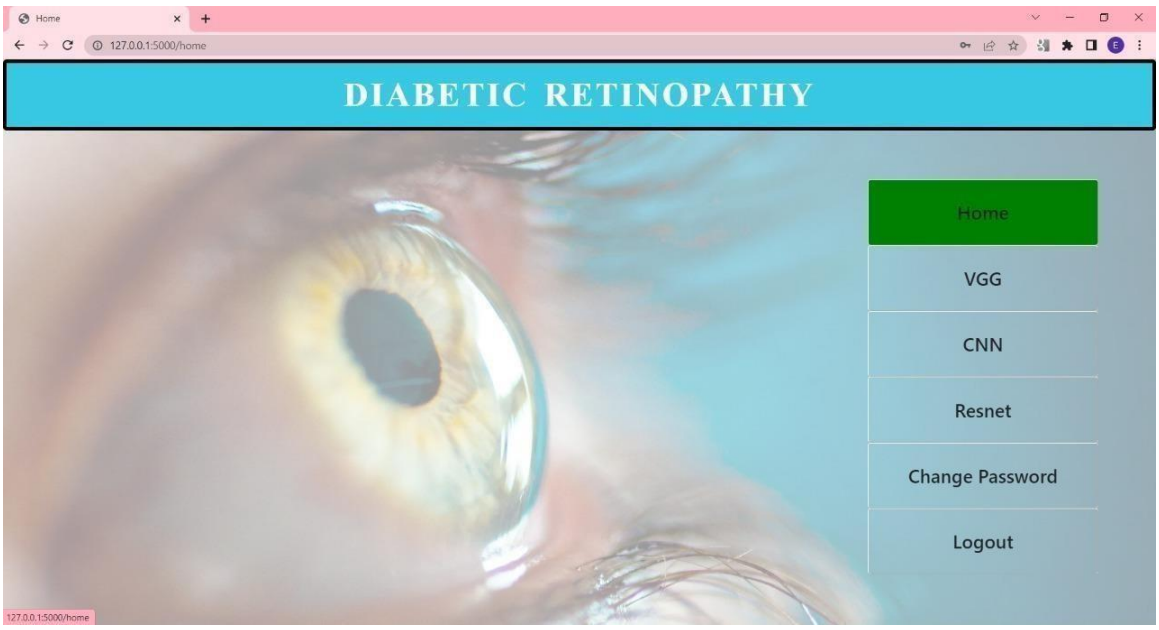


FIGURE 9.2 HOME PAGE

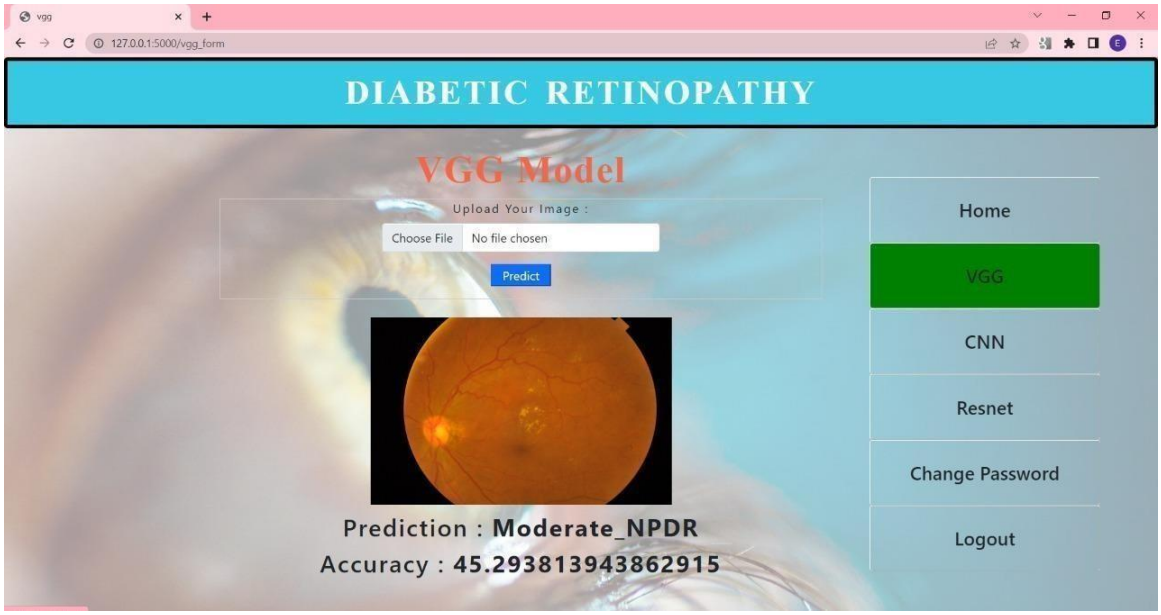


FIGURE 9.3 VGG

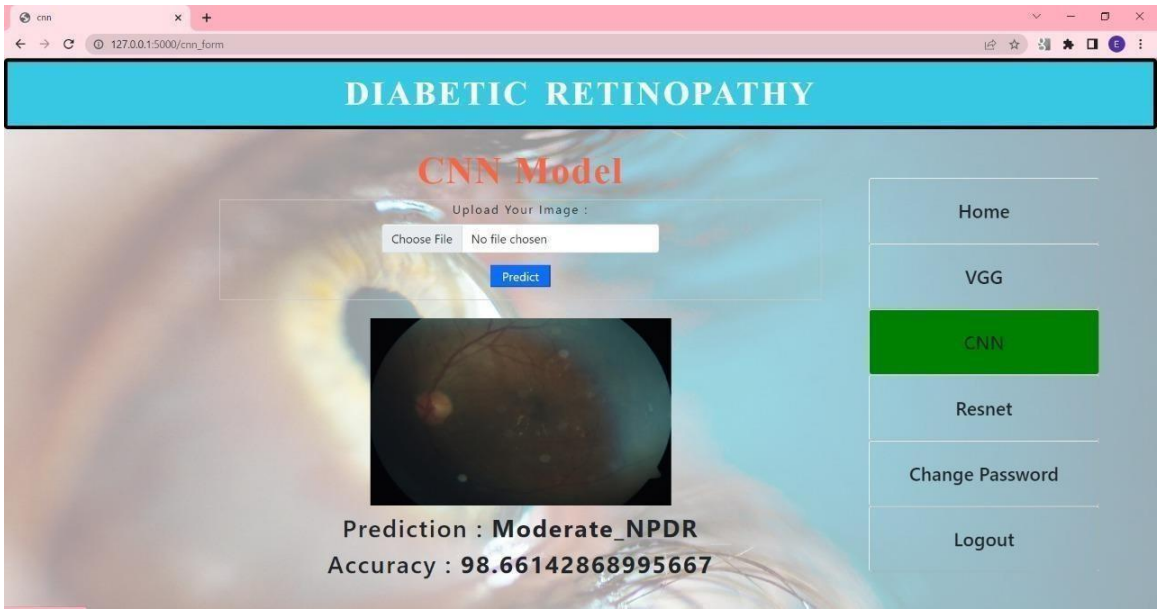


FIGURE 9.4 CNN

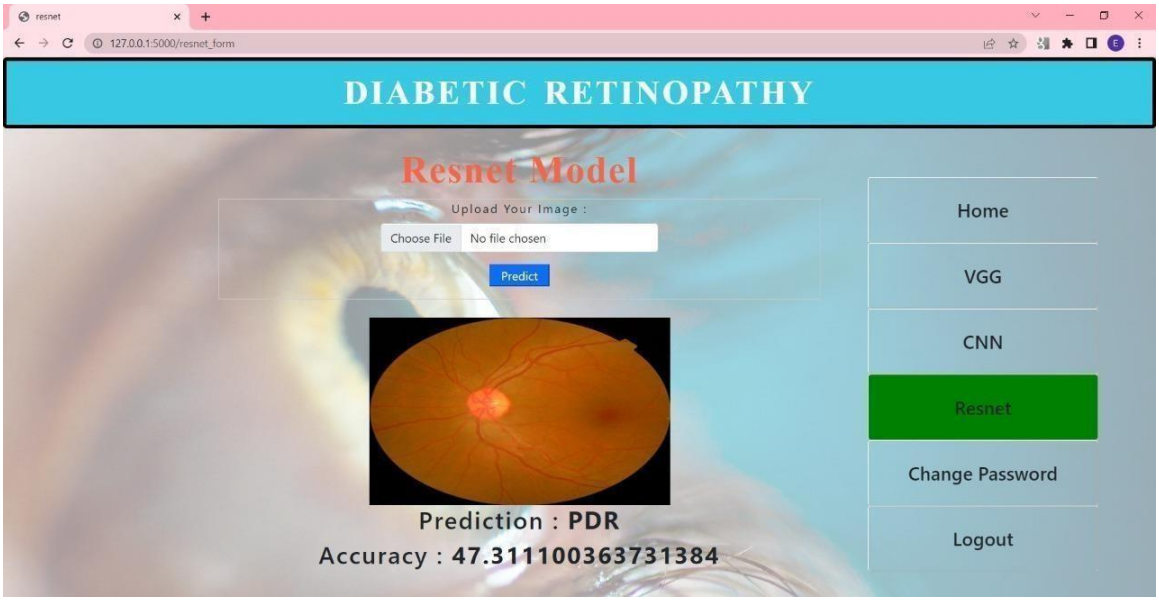


FIGURE 9.5 RESNET15V12

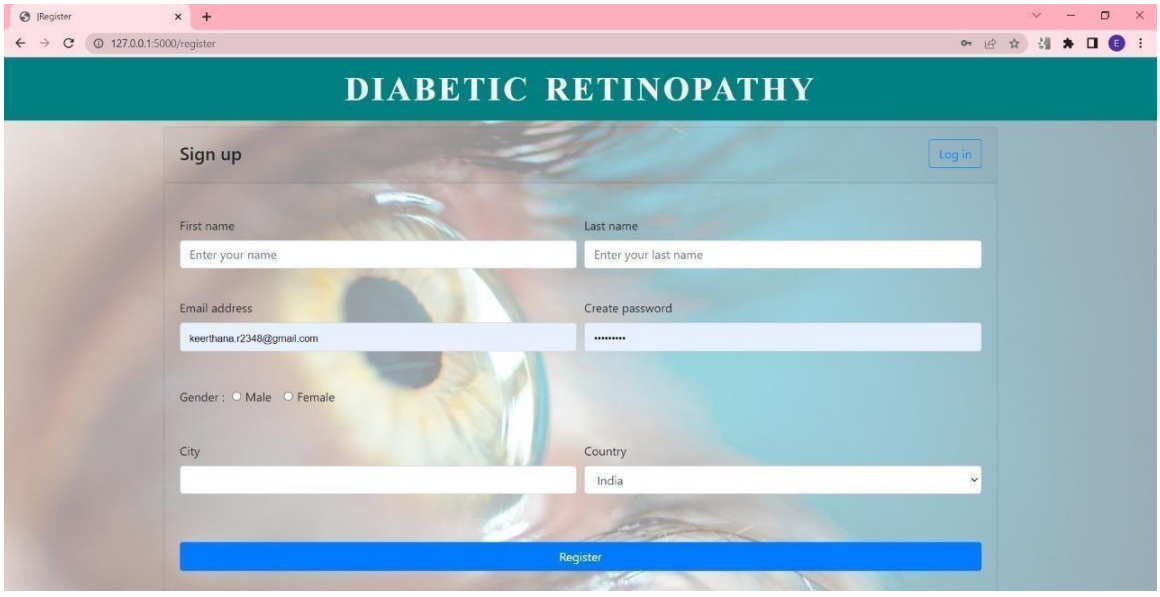


FIGURE 9.6 SIGN UP PAGE

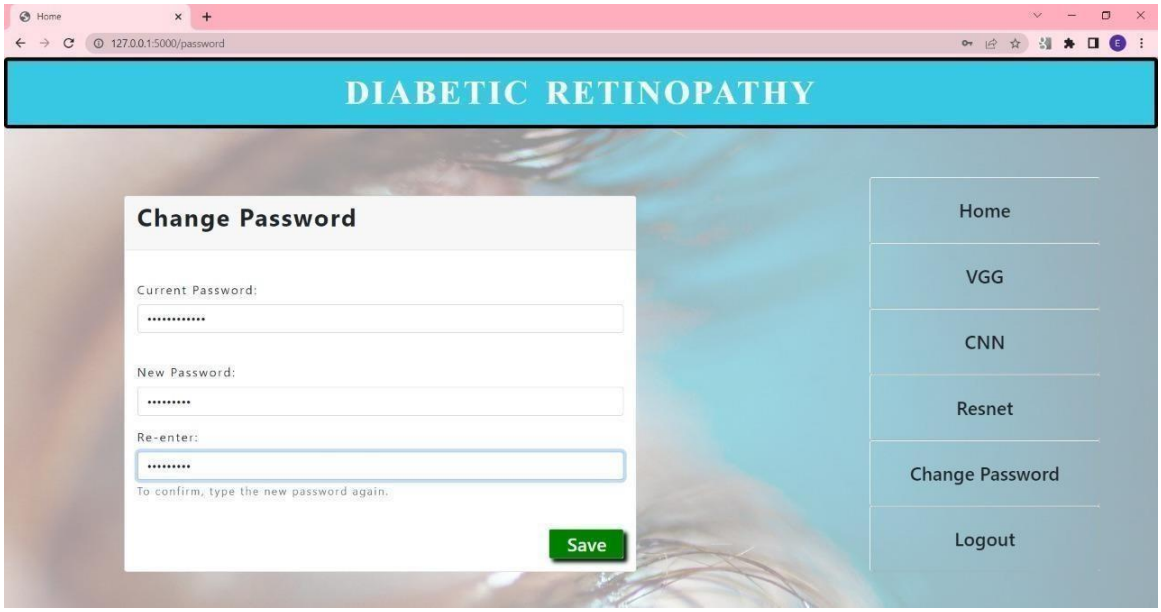


FIGURE 9.7 CHANGE PASSWORD PAGE

CHAPTER 10

CONCLUSION

CHAPTER 10 CONCLUSION

Diabetes was once thought of as a disease of the affluent but it has now reached epidemicproportion in both developed and developing countries. Currently, at least 366 million people worldwide have diabetes, and this number is likely to increase as a result of an agingglobal population.Globally, the number of people with DR will grow from 126.6 million in 2010 to 191.0 million by 2030, and we estimate that the number with vision-threatening diabetic retinopathy (VTDR) will increase from 37.3 million to 56.3 million, if prompt action is not taken. This project is an analysis of a model to identify the severity of DR from Fundus Photographs. Out of 3,662 images, 733 images from the dataset are spitted for validation purpose. Loading the validation images into the models for predicting the label took 188 seconds. We define accuracy as the number of patients with an accurate classification rate. In this five-class problem, the accuracy of CNN model was 96%, ResNet152V2 was 98% and that of VGG-16 was 67%. Thus, ResNet152V2 performs better than the other two models. It is a fact that better and accurate the diagnosis, the more exact will be the treatment plan. So diagnostic measures should aim towards accuracy for an effective treatment regimen. In our study we were able to establish a good accuracy in the diagnosis results

CHAPTER 11

FUTURE WORK

CHAPTER 11 FUTURE WORK

In this Project, have focused on detecting 5 stages of diabetic retinopathy by implementing different deep Learning CNN architectures. In the future, we would focus on detecting different retinal diseases like Retinal tear, Retinal detachment, macular hole, macular degeneration, Retinitis using retinal fundus image. And will try to build a model that will detect all these diseases from a single image along with the stage of the disease. And also try other deep learning models. And perform this classification method using larger dataset of infected eyes. The future scope of project can also be addition of effective WhatsApp assistance.

REFERENCES

[1] Matias Iglicki, MD, PHD, Dinah Zur, MD, and Anat Loewenstein, MD, MHA, are the authors of “Detection of Diabetic Retinopathy Using Deep Learning Analysis” published in retina today 2021 feature.

[2]S. Yu, D. Xiao and Y. Kanagasingam, "Exudate detection for diabetic retinopathy with convolutional neural networks," *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2017, pp. 1744-1747, doi: 10.1109/EMBC.2017.8037180.

[3] Yashal Shakti Kanungo, Bhargav Srinivasan, Dr. Savita Choudhary , “Detecting Diabetic Retinopathy using Deep Learning” , 2017 2nd IEEE International Conference On Recent Trends in Electronics Information & Communication Technology (RTEICT), May 19-20, 2017, India

[4] Kavakiotis, I.; Tsave, O.; Salifoglou, A.; Maglaveras, N.; Vlahavas, I.; Chouvarda, I. Machine learning and data mining methods in diabetes research. Comput. Struct. Biotechnol. J. 2017, 15, 104–116.

[5] Matias Iglicki, MD, PHD, Dinah Zur, MD, and Anat Loewenstein, MD, MHA, “Detection of Diabetic Retinopathy Using Deep Learning Analysis” retina today 2021 feature.

[6] Gulshan, V., Peng, L., Coram, M., Stumpe, M., Wu, D., & Narayanaswamy, A. et al. (2016). Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs. JAMA, 316(22), 2402.

APPENDIX A

CNN

```
def load_training_data():
# Load validation images
labels = os.listdir(train_data_dir)
X_train = np.ndarray((1278, img_rows, img_cols, 3), dtype=np.uint8)
Y_train = np.zeros((1278,), dtype='uint8')
i = 0
print('-'*30)
print('Creating validation images...')
print('-'*30)
j = 0
for label in labels:
    image_names_train = os.listdir(os.path.join(train_data_dir, label))
    if len(image_names_train)>296:
        image_names_train=image_names_train[:296]
    else:
        image_names_train=image_names_train
    total =  len(image_names_train)
    print(label, total)
    for image_name in image_names_train:
        try:
            img = cv2.imread(os.path.join(train_data_dir, label, image_name), 1)
            img = np.array(cv2.resize(img, (img_rows,img_cols)))
            X_train[i] = img
            if len(Y_train)<1480:
                Y_train[i] = j
        except Exception as e:
            pass
        i += 1
    j += 1
print(i)
```

```
print('Loading done.')

print('Transform targets to keras compatible format.')
Y_train = to_categorical(Y_train, num_classes)
return X_train, Y_train
```

APPENDIX B

ResNet152V2

```
def load_training_data():

# Load validation images

labels = os.listdir(train_data_dir)

X_train = np.ndarray((1278, img_rows, img_cols, 3), dtype=np.uint8)

Y_train = np.zeros((1278,), dtype='uint8')

i = 0

print('-'*30)

print('Creating Training images...')

print('-'*30)

j = 0

for label in labels:

    image_names_train = os.listdir(os.path.join(train_data_dir, label))

    if len(image_names_train)>296:

        image_names_train=image_names_train[:296]

    else:

        image_names_train=image_names_train

    total = len(image_names_train)
```

```
print(label, total)

for image_name in image_names_train:

    try:

        img = cv2.imread(os.path.join(train_data_dir, label, image_name), 1)

        img = np.array(cv2.resize(img, (img_rows,img_cols)))

        X_train[i] = img

        if len(Y_train)<1480:

                Y_train[i] = j

            except Exception as e:

                pass

            i += 1

        j += 1

    print(i)

    print('Loading done.')

    print('Transform targets to keras compatible format.')

    Y_train = to_categorical(Y_train, num_classes)

    return X_train, Y_train
```

APPENDIX C

VGG-16

```
def load_training_data():
    # Load validation images
    labels = os.listdir(train_data_dir)
    X_train = np.ndarray((1278, img_rows, img_cols, 3), dtype=np.uint8)
    Y_train = np.zeros((1278,), dtype='uint8')
    i = 0
    print('-'*30)
    print('Creating Training images...')
    print('-'*30)
    j = 0
    for label in labels:
        image_names_train = os.listdir(os.path.join(train_data_dir, label))
        if len(image_names_train)>296:
            image_names_train=image_names_train[:296]
        else:
            image_names_train=image_names_train
        total =  len(image_names_train)
        print(label, total)
        for image_name in image_names_train:
            try:
                img = cv2.imread(os.path.join(train_data_dir, label, image_name), 1)
                img = np.array(cv2.resize(img, (img_rows,img_cols)))
                X_train[i] = img
                if len(Y_train)<1480:
                    Y_train[i] = j
            except Exception as e:
                pass
            i += 1
        j += 1
    print(i)
```

```
print('Loading done.')
print('Transform targets to keras compatible format.')
Y_train = to_categorical(Y_train, num_classes)
return X_train, Y_train
```

GITHUB REPOSITORY LINK

https://github.com/GreeshmaCShekar/Team_10_Diabetic_Retinopathy

PUBLISHED PAPER DETAILS

International Research Journal of Engineering and Technology (IRJET) is a **UGC** recognized, peer reviewed, open access, high Impact Factor, Multidisciplinary journal in English for the enhancement of research in various discipline of Engineering, Science and Technology.

Paper Submission Details:

- Paper Title: Automated Detection of Diabetic Retinopathy Using Deep Learning
- Paper ID: IRJET-V9I5136
- Authors: Greeshma C Shekar, Chaitanya J M Reddy, H V Rakshitha, Harshitha C S,Keerthana R
- Publish Date: 05-May -2022
- Publisher Name: IRJET, published in Volume 9 Issue 5 May 2022
- Website Link: https://www.irjet.net/volume9-issue5
- DOI Link: https://www.irjet.net/archives/V9/i5/IRJET-V9I5136.pdf


e-ISSN: 2395-0056 p-ISSN: 2395-0072
International Research Journal of Engineering and Technology (IRJET)
(An ISO 9001 : 2008 Certified Journal)
Is hereby awarding this certificate to
Greeshma C Shekar
In recognition the publication of the manuscript entitled
Automated Detection of Diabetic Retinopathy using Deep Learning
published in our Journal Volume 9 Issue 5 May 2022
Impact Factor : 7.529
www.irjet.net
Editor in Chief
E-mail : editor@irjet.net


e-ISSN: 2395-0056 p-ISSN: 2395-0072
International Research Journal of Engineering and Technology (IRJET)
(An ISO 9001 : 2008 Certified Journal)
Is hereby awarding this certificate to
HV Rakshitha
In recognition the publication of the manuscript entitled
Automated Detection of Diabetic Retinopathy using Deep Learning
published in our Journal Volume 9 Issue 5 May 2022
Impact Factor : 7.529
www.irjet.net
Editor in Chief
E-mail : editor@irjet.net

e-ISSN: 2395-0056 p-ISSN: 2395-0072

International Research Journal of Engineering and Technology (IRJET)

(An ISO 9001 : 2008 Certified Journal)

Is hereby awarding this certificate to

Harshitha CS

In recognition the publication of the manuscript entitled

Automated Detection of Diabetic Retinopathy using Deep Learning

published in our Journal Volume 9 Issue 5 May 2022

Impact Factor : 7.529 www.irjet.net

Editor in Chief

E-mail : editor@irjet.net

e-ISSN: 2395-0056 p-ISSN: 2395-0072

International Research Journal of Engineering and Technology (IRJET)

(An ISO 9001 : 2008 Certified Journal)

Is hereby awarding this certificate to

Keerthana R

In recognition the publication of the manuscript entitled

Automated Detection of Diabetic Retinopathy using Deep Learning

published in our Journal Volume 9 Issue 5 May 2022

Impact Factor : 7.529 www.irjet.net

Editor in Chief

E-mail : editor@irjet.net

Automated Detection of Diabetic Retinopathy Using Deep Learning

Greeshma C Shekar[1], Chaitanya J M Reddy[2], HV Rakshitha[3], Harshitha CS[4], Keerthana R[5]

[1,2,3,4,5]*Department of Computer Science and Engineering, Dayananda Sagar University, Bangalore, Karnataka, India*

Abstract - *Diabetic retinopathy is one of the prevalent causes of blindness among working-age adults. Diabetic retinopathy or DR is an ailment because of diabetes mellitus that can harm the patient image retina and also cause blood spills It is the fastest growing cause of blindness. We have used Deep learning classification techniques, Convolutional Neural Network (CNN), pre-trained VGG-16, ResNet to detect the severity level of Diabetic Retinopathy from the color fundus image. Fundus photography technique is used to take these photographs.*

Key Words*: *Diabetic Retinopathy, Convolutional Neural Network, VGG-16 and ResNet

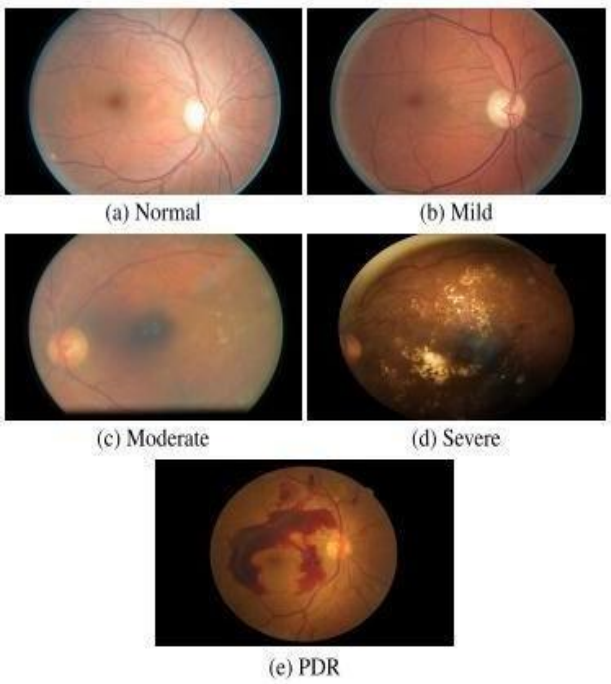
1. INTRODUCTION

Diabetic retinopathy is a medical complication that is caused by the damage to the blood vessels of the light-sensitive tissue which is present at the back of the eye, retina, which can gradually lead to complete blindness and various other eye problems depending on the severity of Diabetic Retinopathy.

It is observed that 40% – 45% of diabetic patients are likely to have DR in their life, but due to lack of knowledge and delayed diagnosis, the condition escalates quickly.

Diabetes was once thought of as a disease of the affluent but it's now reached epidemic proportion in both developed and developing countries. Currently, a minimum of 366 million people worldwide has diabetes, and this number is probably going to extend as a results of an aging global population

Globally, the quantity of individuals with DR will grow from 126.6 million in 2010 to 191.0 million by 2030, and that we estimate that the quantity with vision-threatening diabetic retinopathy (VTDR) will increase from 37.3 million to 56.3 million, if prompt action isn't taken.



1.1 METHOD AND METHODOLOGY

1.2 DATA SET

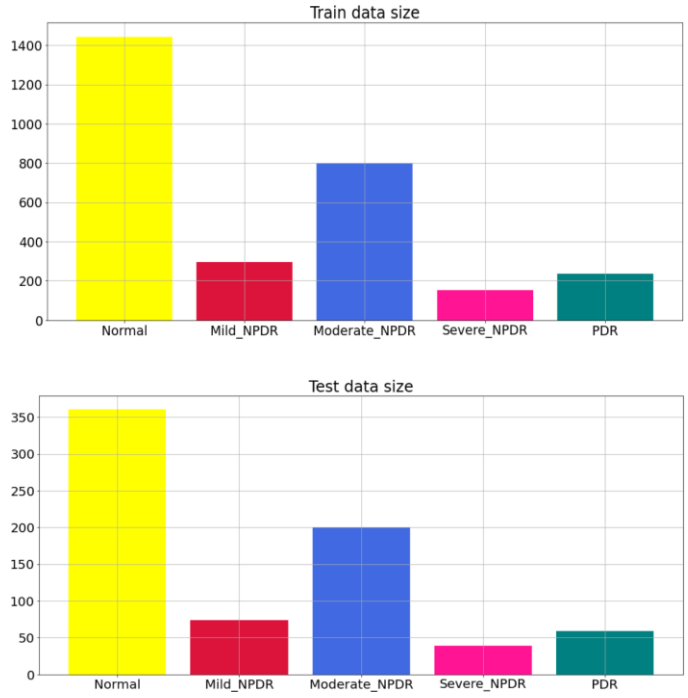
We have used the data collected by the Asia-Pacific TeleOphthalmology Society (APTOS) available on the Kaggle platform.

The data set consists of 3,662 color fundus images. We have used the dataset of such images for the training and testing of our model. Each image in the dataset has been assigned an integral value on the scale of 0 to 4 according to the severity of the disease by a professionally trained clinician as shown in Table-1.

Class	Name	No. of images
0	Normal	1805
1	Mild NPDR	370
2	Moderate NPDR	999
3	Severe NPDR	193
4	PDR	295

Table -1

In order to induce a more balanced date, we have split the dataset into 80:20 for training and validation purpose. Totally for training purpose we have considered 2,929 color fundus images and 733 color fundus images for validation.



0 – Normal: The person is not suffering from Diabetic Retinopathy.

1 – Mild Non-Proliferative DR (Mild_NPDR): Within the Retina's minute blood vessels, small areas of balloon like inflammations.

2 – Moderate Non-Proliferative DR (Moderate_NPDR): The blood vessels that sustain the retina are blocked at this stage. Within the retina, there might also be haemorrhages.

3 – Severe Non-Proliferative DR (Severe_NPDR): More blood vessels are blocked in this particular stage, denying several areas of retina of blood supply. The number of haemorrhages in the retina also increases drastically.

4 – Proliferative DR (PDR): New and abnormal blood vessels developed on the surface of retina. These new blood vessels are delicate and have the tendency to bleed, causing vision threatening haemorrhages to fill the eye. They can also turn into connective tissue which will contract over time, causing the retina to detach and cause blindness.

2. PREPROCESSING

We are resizing the images for pre-processing the data before providing it to the model first, keeping in view the aspect ratio to 256 × 256. It helps us to avoid features loss from images. The dataset is distributed into training, validation sets with a ratio of 80% and 20% respectively.

CNN ARCHITECTURE:

Convolutional Neural Network, is commonly used to breaking down of visual imagery and thus used for retinal images. A Convolutional Neural Network or CNN is a general multilayered neural framework with an outstanding plan to perceive complex features in the data. The pre-processing required in a CNN is much lower as compared to other classification algorithms. CNN architecture consists of these main layers: convolutional layer, pooling layer, Dropout layer and fully-connected layer.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 256, 256, 64)	1792
max_pooling2d (MaxPooling2D)	(None, 128, 128, 64)	0
conv2d_1 (Conv2D)	(None, 128, 128, 64)	36928
max_pooling2d_1 (MaxPooling2D)	(None, 64, 64, 64)	0
conv2d_2 (Conv2D)	(None, 64, 64, 128)	73856
max_pooling2d_2 (MaxPooling2D)	(None, 32, 32, 128)	0
conv2d_3 (Conv2D)	(None, 32, 32, 128)	147584
max_pooling2d_3 (MaxPooling2D)	(None, 16, 16, 128)	0
dropout (Dropout)	(None, 16, 16, 128)	0
flatten (Flatten)	(None, 32768)	0
dense (Dense)	(None, 1024)	33555456
dense_1 (Dense)	(None, 5)	5125

Total params: 33,820,741
Trainable params: 33,820,741
Non-trainable params: 0

We have built the sequential CNN model. Firstly, Conv2d layer with parameters: 64 filters and kernel of size (3,3) which is a 2-tuple specifying width and height of the 2D convolution window.

The input shape parameter is passed with train's shape and with padding value to be the same to preserve the spatial dimensions along with activation parameter to be Relu.

Max pooling is considered as the pooling layer, which down-samples the resulting feature maps and increases the receptive field on the filters [2,2].

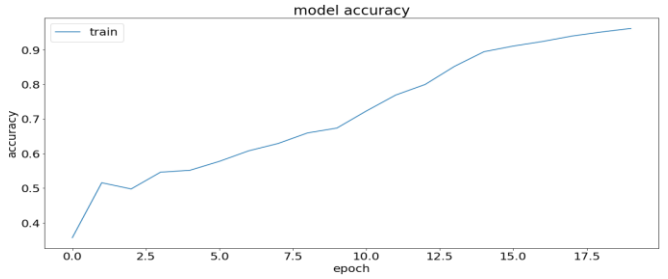
These 2 layers are re-layered in the same manner once more.

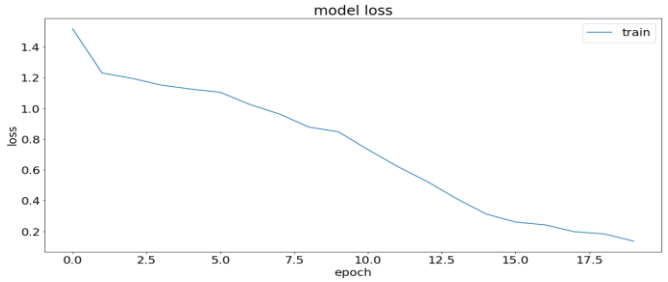
Now a conv2d layer with parameters of 128 filters and (3,3) kernel with padding and activation is to be the same and Relu respectively.

Then a pooling layer is added wherein max pooling is done. These 2 layers are re-layered in the same manner twice.

To maintain the effectivity of the data and avoid overfitting, a Dropout Layer (DL) is added succeeding to the fully-connected layer. Dropout usually randomly deactivates a fraction of the units or connections for example 50%, in a network on each training iteration.

After dropout and flattening the model, a dense layer is added with 1024 units and activation to be Relu.





CNN

VGG-16 ARCHITECTURE:

Model: "model"

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 256, 256, 3)]	0
block1_conv1 (Conv2D)	(None, 256, 256, 64)	1792
block1_conv2 (Conv2D)	(None, 256, 256, 64)	36928
block1_pool (MaxPooling2D)	(None, 128, 128, 64)	0
block2_conv1 (Conv2D)	(None, 128, 128, 128)	73856
block2_conv2 (Conv2D)	(None, 128, 128, 128)	147584
block2_pool (MaxPooling2D)	(None, 64, 64, 128)	0
block3_conv1 (Conv2D)	(None, 64, 64, 256)	295168
block3_conv2 (Conv2D)	(None, 64, 64, 256)	590080
block3_conv3 (Conv2D)	(None, 64, 64, 256)	590080
block3_pool (MaxPooling2D)	(None, 32, 32, 256)	0
block4_conv1 (Conv2D)	(None, 32, 32, 512)	1180160
block4_conv2 (Conv2D)	(None, 32, 32, 512)	2359808
block4_conv3 (Conv2D)	(None, 32, 32, 512)	2359808
block4_pool (MaxPooling2D)	(None, 16, 16, 512)	0
block5_conv1 (Conv2D)	(None, 16, 16, 512)	2359808
block5_conv2 (Conv2D)	(None, 16, 16, 512)	2359808
block5_conv3 (Conv2D)	(None, 16, 16, 512)	2359808
block5_pool (MaxPooling2D)	(None, 8, 8, 512)	0
global_max_pooling2d (GlobalMaxPooling2D)	(None, 512)	0
dense (Dense)	(None, 1024)	525312
dense_1 (Dense)	(None, 5)	5125

Total params: 15,245,125
Trainable params: 530,437
Non-trainable params: 14,714,688

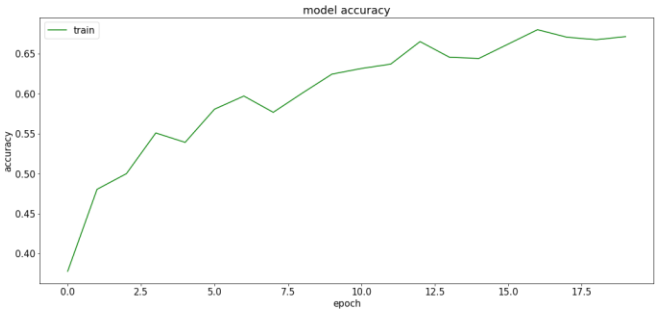
VGG16 is one of the best vision model architectures present. VGG16 is having a large number of hyper-parameters, focused on having convolution layers of 3x3 filter with stride 1 and has same padding and max pool layer of 2x2 filter of stride 2.

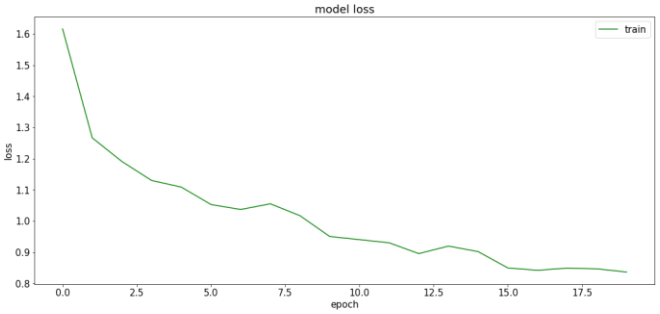
This arrangement of convolution and max pool layers is consistently followed throughout the entire architecture.

At the end, it has 2 fully connected layers followed by a SoftMax for output.

VGG16 has 16 layers that have weights.

We downloaded the VGG-16 model from TensorFlow Keras. Then added fully connected layer, dense layer with 1024 units and activation to be Relu.





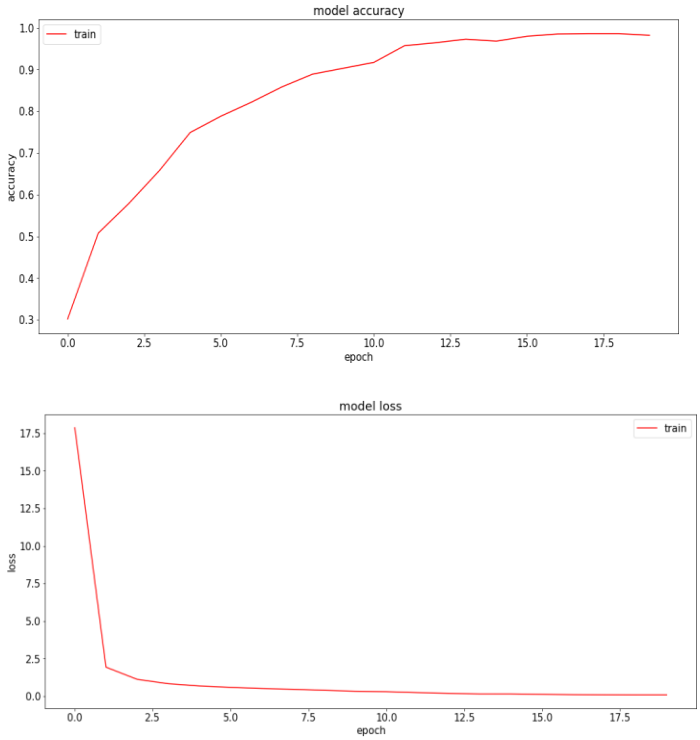
VGG16

ResNet152V2 ARCHITECTURE:

ResNet-152V2, a convolutional neural network has 101 layers deep. we will load a pre-trained version of the network, trained on over 1,000,000 images.

The pre-trained network can classify images into 1000 object categories, as a result, the network has learned rich feature representations for a wider range of images. The network has a picture input size of 224-by-224.

We downloaded the ResNet152V2 model from TensorFlow Keras. Then added fully connected layer, dense layer with 1024 units and activation to be Relu.



ResNet152V2

For all the 3 models, epoch is defined to be once all images are processed once individually of forward and backward to the network. 1 epoch is counted to be:

(Number of iterations * batch size) / total number of images present in training.

While training the models, we checked for 20 epochs, with batch size 64. Each epoch as observed in the above graphs, the loss decreased simultaneously the accuracy accuracy increased.

GUI:

A web application has been developed, where the color fundus image can be uploaded as the input to the models by the user who can login with credentials.

The respective result according to the stage of Diabetic Retinopathy will be displayed along with accuracy. The email of the result shall be sent to the patient.

The database consisting of the patient's records are maintained using MySQL

3. IMPLEMENTATION

$$\text{True Positive Rate} = \frac{TP}{TP + FN}$$

$$\text{True Negative Rate} = \frac{TN}{TN + FP}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

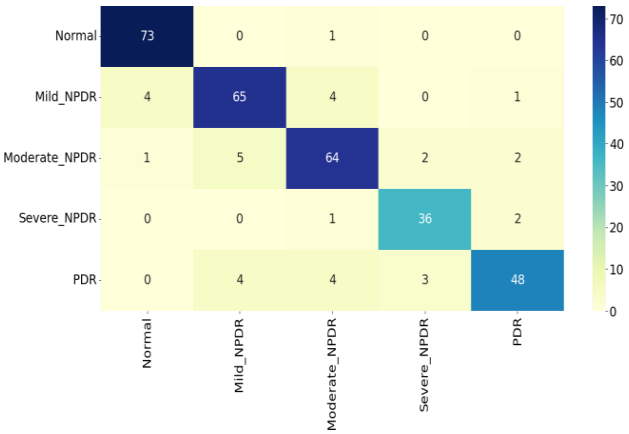
The true positive rate also referred to as hit rate or recall of a classifier is estimated by dividing the correctly classified positives by the full positive count.

The false positive rate of the classifier is estimated by dividing the incorrectly classified negatives by the overall negatives.

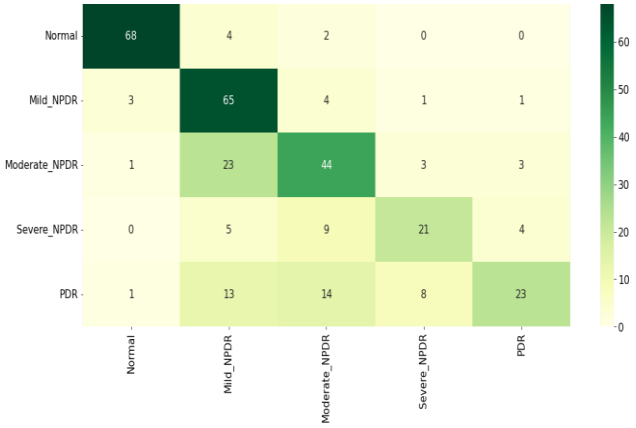
The accuracy of a classifier is estimated by dividing the full correctly classified positives and negatives by the overall number of samples.

Precision is one indicator which tells the quality of positive predictions made by the models.

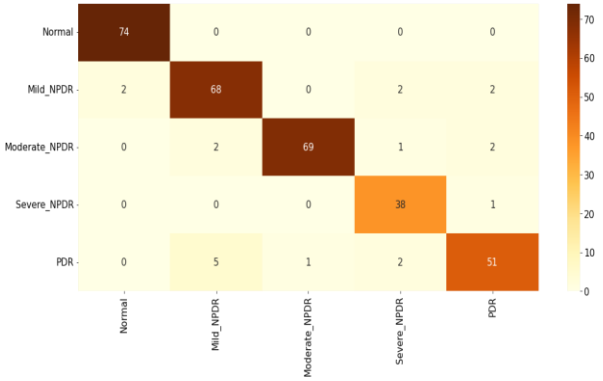
Recall is the ratio of True Positive by the sum of True Positive and False Negative.



CNN



VGG16



ResNet152V2

4. CONCLUSION

Out of 3,662 images, 733 images from the dataset are spitted for validation purpose. Loading the validation images into the models for predicting the label took 188 seconds.

We define accuracy as the number of patients with an accurate classification rate.

In this five-class problem, the accuracy of CNN model was 96%, ResNet152V2 was 98% and that of VGG-16 was 67%. Thus, ResNet152V2 performs better than the other two models.

	precision	recall	f1-score	support
Normal	0.94	0.99	0.96	74
Mild_NPDR	0.88	0.88	0.88	74
Moderate_NPDR	0.86	0.86	0.86	74
Severe_NPDR	0.88	0.92	0.90	39
PDR	0.91	0.81	0.86	59

CNN

	precision	recall	f1-score
Normal	0.93	0.92	0.93
Mild_NPDR	0.59	0.88	0.71
Moderate_NPDR	0.60	0.59	0.60
Severe_NPDR	0.64	0.54	0.58
PDR	0.74	0.39	0.51

VGG16

	precision	recall	f1-score	support
Normal	0.97	1.00	0.99	74
Mild_NPDR	0.91	0.92	0.91	74
Moderate_NPDR	0.99	0.93	0.96	74
Severe_NPDR	0.88	0.97	0.93	39
PDR	0.91	0.86	0.89	59

ResNet152V2

REFERENCES

[1] Yashal Shakti Kanungo, Bhargav Srinivasan, Dr. Savita Choudhary, "Detecting Diabetic Retinopathy using Deep Learning", 2017 2nd IEEE International Conference On Recent Trends in Electronics Information & Communication Technology (RTEICT), May 19-20, 2017, India

[2] Kavakiotis, I.; Tsave, O.; Salifoglou, A.; Maglaveras, N.; Vlahavas, I.; Chouvarda, I. Machine learning and data mining methods in diabetes research. Comput. Struct. Biotechnol. J. 2017, 15, 104–116.

[3] Matias Iglicki, MD, PHD, Dinah Zur, MD, and Anat Loewenstein, MD, MHA, "Detection of Diabetic Retinopathy Using Deep Learning Analysis" retina today 2021 feature.

[4] Shorav Suriyal, Christopher Druzgalski, Kumar Gautam, "Mobile Assisted Diabetic Retinopathy Detection using Deep Neural Network," MARCH 19–24, 2018 ISBN: 978-1-5386-5475-0 IEEE CATALOG NUMBER: CFP1818GART 978-1-5386-5475-0/18/$31.00 ©2018 IEEE

[5] Amol Prataprao Bhatkar, Dr. G.U. Kharat, "Detection of Diabetic Retinopathy in Retinal Images using MLP classifier" 978-1-4673-9692-9/15 $31.00 © 2015 IEEE DOI 10.1109/iNIS.2015.30

[6] Mohamed Chetoui, Moulay A. Akhloufi, Mustapha Kardouchi, "Diabetic Retinopathy Detection Using Machine Learning and Texture Features," 978-1-5386-2410-4/18/$31.00 ©2018 IEEE

[7] XIANGLONG ZENG, HAIQUAN CHEN, YUAN LUO AND WENBIN YE, "Automated Diabetic Retinopathy Detection Based on Binocular Siamese-like

Convolutional Neural Network" 2169-3536 (c) 2018 IEEE

[8] Z. A. Omar, M. Hanafi, S. Mashohor, N. F. M. Mahfudz, M. Muna'im, "Automatic diabetic retinopathy" 978-1- 5386-0383-3/17/$31.00 ©2017 IEEE

[9] Yashal Shakti Kanungo, Bhargav Srinivasan, Dr. Savita Choudhary, "Detecting Diabetic Retinopathy using Deep Learning" 978-1-5090-3704-9/17/$31.00 © 2017 IEEE

[10] Deepthi K Prasad, Vibha L, Venugopal K R, "Early Detection of Diabetic Retinopathy from Digital Retinal Fundus Images" 978-1-4673-6670-0/15/$31.00 ©2015 IEEE

[11] Enrique V. Carrera, Andr´es Gonz´alez, Ricardo Carrera, "Automated detection of diabetic retinopathy using SVM" 978-1-5090-6363-5/17/$31.00 _c 2017 IEEE

[12] Ratul Ghosh, Kuntal Ghosh, Sanjit Maitra, "Automatic Detection and Classification of Diabetic Retinopathy stages using CNN" 978-1-5090-2797-2/17/$31.00 ©2017 IEEE

BIOGRAPHIES

Miss. Greeshma C Shekar is pursuing Bachelor of Technology degree in department of Computer Science and Engineering from Dayananda Sagar University, located in Bangalore, Karnataka, India. She has published papers entitled," Birds Audio Detection using Convolutional Neural Network and Transfer Learning" in IRJET Volume-8 Issue-9, September 2021 and "Blockchain and Cryptocurrency: The world of Blockchain and Cryptocurrency" in IJCSE Volume-9 Issue-7, July 2021. She is currently in her final year of Engineering and will be graduating from Dayananda Sagar University in the year 2022.

Miss. Keerthana R is pursuing Bachelor of Technology degree in department of Computer Science and Engineering from Dayananda Sagar University, located in Bangalore, Karnataka, India. She is currently in her final year of Engineering and will be graduating from Dayananda Sagar University in the year 2022.

Miss. HV Rakshitha is pursuing Bachelor of Technology degree in department of Computer Science and Engineering from Dayananda Sagar University, located in Bangalore, Karnataka, India. She is currently in her final year of Engineering and will be graduating from Dayananda Sagar University in the year 2022.

Miss. Chaitanya J M Reddy is pursuing Bachelor of Technology degree in department of Computer Science and Engineering from Dayananda Sagar University, located in Bangalore, Karnataka, India. She is currently in her final year of Engineering and will be graduating from Dayananda Sagar University in the year 2022.

Miss. Harshitha CS is pursuing Bachelor of Technology degree in department of Computer Science and Engineering from Dayananda Sagar University, located in Bangalore, Karnataka, India. She is currently in her final year of Engineering and will begraduating from Dayananda Sagar University in the year 2022.