

PROJECT REPORT
ON
A MACHINE LEARNING APPROACH TO DETECT DIABETIC
RETINOPATHY



DAFFODIL INTERNATIONAL UNIVERSITY
DHAKA, BANGLADESH
OCTOBER 2020

A MACHINE LEARNING APPROACH TO DETECT DIABETIC RETINOPATHY

BY

PRATIM SAHA

ID: 141-15-105

This Report Presented in Partial Fulfillment of the Requirements for the
Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Md. Nazmul Hoq

Lecturer & Associate Head

Department of CSE

Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

OCTOBER 2020

APPROVAL

This Project titled “**A machine learning approach to detect diabetic retinopathy**”, submitted by Pratim Saha, ID: 141-15-105 to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on October 8, 2020.

BOARD OF EXAMINERS

Dr. Syed Akhter Hossain
Professor and Head

Department of CSE
Faculty of Science & Information Technology
Daffodil International University

Chairman

Mr. Narayan Ranjan Chakraborty
Assistant Professor

Department of CSE
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Md. Nazmul Hoq
Lecturer and Associate Head

Department of CSE
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Dr. Mohammad Shorif Uddin
Professor and Chairman

Department of CSE
Jahangirnagar University

External Examiner

DECLARATION

I hereby declare that, this project has been done by me under the supervision of **Md. Nazmul Hoq, Lecturer & Associate Head, Department of CSE** Daffodil International University. I also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:



Md. Nazmul Hoq
Lecturer & Associate Head
Department of CSE
Daffodil International University

Submitted by:



Pratim Saha
ID: 141-15-105
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First I express my heartiest thanks and gratefulness to almighty God for His divine blessing makes me possible to complete the final year project successfully.

I am really grateful and wish my profound indebtedness to **Md. Nazmul Hoq, Lecturer & Associate Head**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of my supervisor in the field of “*Machine learning*” to carry out this project. His endless patience ,scholarly guidance ,continual encouragement , constant and energetic supervision, constructive criticism , valuable advice ,reading many inferior draft and correcting them at all stage have made it possible to complete this project.

I would like to express my heartiest gratitude to **Dr. Syed Akhter Hossain, Professor, and Head**, Department of CSE, for his kind help to finish our project and also to other faculty members and the staffs of CSE department of Daffodil International University.

I would like to thank my entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

ABSTRACT

Diabetic Retinopathy is an eye disease that affects mainly the retina. It is the consequence of long-standing Diabetes which results as blood leaks from the retinal blood vessels onto the retina. It is one of the most leading causes of blindness in today's world. Therefore early diagnosis and treatment are a must to save people's eyesight. It is very difficult to identify the disease at each earliest stage since the symptoms do not develop until there is significant damage to the retina. Therefore, regular monitoring of the retina is a must to prevent the disease. A significant advancement in the field of biomedicine has made it possible for early detection and diagnosis of this disease. In this paper, I have proposed a machine learning approach using **DenseNet-121** to detect five stages of Diabetic retinopathy namely no-DR, mild-DR, moderate-DR, severe DR, and proliferative-DR with an accuracy of 94%. For the experiment, I have also used some other models for example Inception V3, VGG 19, DenseNet-201, XceptionNet, and traditional CNN to find the best model for my project. I have used 3113 retinal fundus images for the training purpose and 549 images for the validation purpose. A web application was developed to easily check for disease simply by uploading the retinal fundus image. Flask web framework was used for the deployment of the model.

TABLE OF CONTENTS

CONTENTS	PAGE
Title Page	i
Approval	ii
Board of examiners	iii
Declaration	iv
Acknowledgements	v
Abstract	vi
List of Figures	x
List of Tables	xi
CHAPTER	
CHAPTER 1: Introduction	1-4
1.1 Introduction	1-2
1.2 Motivation	3
1.3 Rationale of the Study	3
1.4 Research Questions	4
1.5 Expected Output	4
1.6 Project Management and Finance	4

1.7 Report Layout	4
CHAPTER 2: Background	5-8
2.1 Terminology	5
2.2 Related Works	5-6
2.3 Comparative Analysis and Summary	7
2.3 Scope of the Problem	7
2.4 Challenges	8
CHAPTER 3: Research Methodology	9-21
3.1 Research Subject and Instrumentation	9
3.2 Data Collection Procedure	9
3.3 Statistical Analysis	10
3.4 Proposed Methodology	10-19
3.5 Implementation Requirements	20-21
CHAPTER 4: Experimental Results and Discussion	22-28
4.1 Experimental Setup	22
4.2 Experimental Results & Analysis	23-27
4.3 Discussion	27-28
CHAPTER 5: Impact on Society, Environment and Sustainability	29-30
5.1 Impact on Society	29
5.2 Impact on Environment	29
5.3 Ethical Aspects	29
5.4 Sustainability Plan	30

CHAPTER 6: Summary, Conclusion, Recommendation and Implication for Future Research	31-32
6.1 Summary of the Study	31
6.2 Conclusions	31
6.3 Implication for Future Study	32
REFERENCES	33-35
APPENDICES	36
PLAGIARISM RESULT	37-39

LIST OF FIGURES

FIGURES	PAGE NO
Figure 1.1.1: Typical color fundus image with its five component	2
Figure 1.1.2: Different Stages of Diabetic Retinopathy	2
Figure 3.4.1: Block Diagram of Methodology	11
Figure 3.4.2: Typical CNN Architecture	13
Figure 3.4.3: XceptionNet Architecture	14
Figure 3.4.4: Inception V3 Architecture	15
Figure 3.4.5: VGG 19 Architecture	17
Figure 3.4.6: DenseNet Architecture	19
Figure 3.4.7: Page to Upload Image	19
Figure 3.4.8: Web Page With Predicted Diabetic Retinopathy	20
Figure 4.2.1: (a) VGG19 model accuracy; (b) VGG19 model loss	24
Figure 4.2.2: (a) Inception model accuracy; (b) Inception model loss	24
Figure 4.2.3: (a) XceptionNet model accuracy; (b) XceptionNet model loss	25
Figure 4.2.4: (a) DenseNet-201 model accuracy; (b) DenseNet-201 model loss	25
Figure 4.2.5: (a) DenseNet-121 model accuracy; (b) DenseNet-121 model loss	26
Figure 4.2.6: Accuracy of different models	27

LIST OF TABLES

TABLES	PAGE NO
Table 3.3.1: Statistical analysis of images of each class of Diabetic Retinopathy	10
Table 4.2.1: Experimental Result of different neural network model	23
Table 4.3.1: Class-wise performance of DenseNet-121	28

CHAPTER 1

Introduction

1.1 Introduction

Diabetes is caused by impairment of the body tissue to process blood glucose resulting in an increase of plasma glucose. It affects all the vital organs of the body especially the Kidney, Brain, and Eye resulting in diabetic nephropathy, neuropathy, and retinopathy respectively. In 2010, 0.8 million people globally were declared blind and another 3.7 million were visually impaired because of diabetic retinopathy. At present, a minimum of 366 million people are suffering from diabetes around the world and approximately 41.6% of them are suffering from any form of DR. It is estimated that by the year 2030 total number of people with DR will be 191 million. [23] Diabetic retinopathy occurs when the blood supplying vessels of the retina become unimpaired. The consequence of damage to the small blood vessels and nerves of the retina are diabetic retinopathy. The earliest changes leading to diabetic retinopathy include narrowing of the retinal arteries associated with reduced retinal blood flow; malfunction of the inner retina neurons, accompanied later by improvements in the activity of the outer retina, associated with gradual changes in visual performance; blood-retina barrier malfunction that protects the retina from many substances Up to 80% of people who have had diabetes for 20 years or longer have diabetic retinopathy. A patient can only detect it at its advanced stage when treatment is very complicated and people often lose their eyesight. With proper eye care and surveillance, at least 90 percent of new cases will be minimized. [22] Doctors suggest every diabetic patient consult an ophthalmologist every six months. But rural poor people cannot afford to go to a doctor every six months. In a fundus picture, there are primarily five anatomical components, the recognition of which is essential for diabetic retinopathy diagnosis, namely optic disc, blood vessel, exudate, microaneurysm, and hemorrhage. In Figure 1.1.1, a standard fundus picture is shown with these components labelled.

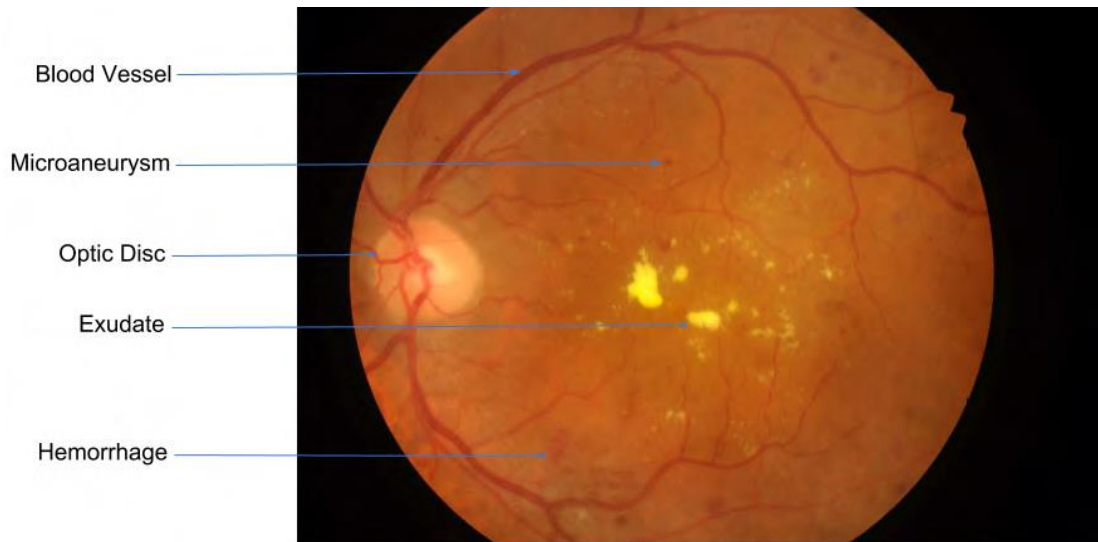


Figure 1.1.1: Typical color fundus image with its five component [21]

There are several features of Diabetic retinopathy which show different manifestation at different stages. It is very time-consuming for any practitioner to detect different stages of Diabetic retinopathy. But with the advancement of technology, it is possible to detect human diseases within seconds. Many machine learning approaches have been used to detect Diabetic retinopathy such as SVM, K-NN, and traditional CNN.

In this project, I have used a transfer learning approach which is very effective for classifying images especially when there is a limited number of training samples. Figure 1.1.2 illustrates different stages of Diabetic Retinopathy.

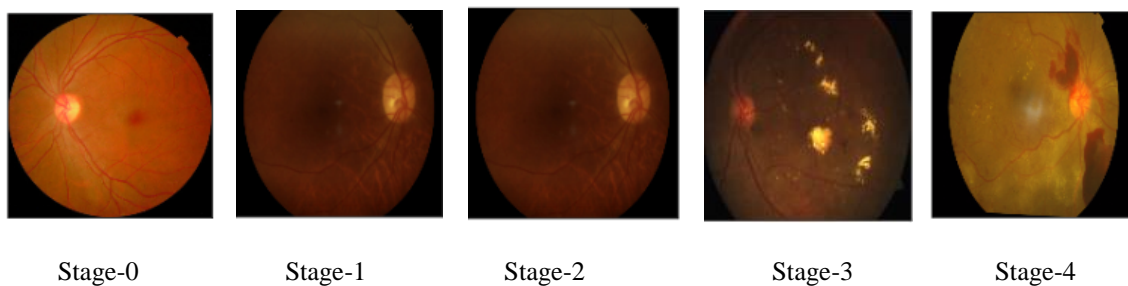


Figure-1.1.2: Different stages of Diabetic Retinopathy

1.2 Motivation

DR was found to be the fifth most prevalent cause of preventable or curable blindness during the 1990-2010 period. [24] In order to reduce permanent damage to vision or blindness, DR care has to be begun at an early stage. For that, early detection is a prerequisite. At present, relative to the large amount of photographs that need to be diagnosed each year, there are very few ophthalmologists. By manual inspection of photographs and detection of lesions, diagnosis is carried out. This job is, sadly, incredibly labor-intensive, time-consuming and vulnerable to human error. An automatic method of detection will come to the assistance of an ophthalmologist and thereby improve the detection efficiency. In comparison, it has been observed that there is no evidence of retinopathy in a significant fraction of screened images. [25] Diabetes has become a global phenomenon and people of third world countries are the worst sufferers of this disease. They have a shortage of money, skilled practitioners. They can't travel a long way to test their eyesight every six months.

This project will let the practitioners in the rural areas to test for Diabetic Retinopathy within seconds and at a very low cost. Successful implementation of this project will save the eyesight of a lot of people.

1.3 Rationale of the Study

A lot of researches have already been done on Diabetic Retinopathy detection with the Caucasian people's retinal image. Although there is a significant level of accuracy has been achieved by different machine learning algorithms but none of them work as good for the South Asian people as the Caucasians. So there is a significant amount of misclassification for the people of this region.

The goal of this project is to focus on the diagnosis of the people belonging to the Indian sub-continent. The present project work has been done with the dataset collected from different areas of India. So the people of the Indian sub-continent will be benefited much from the developed project.

1.4 Research Questions

- a. Will the research play important role in detecting Diabetic Retinopathy? Does the project is capable of detecting different stages of Diabetic Retinopathy or not?
- b. Does the project provide the correct result? If the project can successfully detect different stages of Diabetic Retinopathy what is the accuracy rate of the proposed model?

1.5 Expected Output

In this project, the user can provide any sized retinal fundus image, and the model will automatically convert it into its desired size and display the stage of Diabetic Retinopathy.

1.6 Project Management and Finance

The project was arranged under the direct supervision of my project supervisor, and this project is self-financed. Required materials were partially collected from the internet and some resources were provided by the project supervisor Md. Nazmul Hoq.

1.7 Report Layout

This report is organized as follows:

Chapter 2 will contain a research background that is related to work, research summary, and challenges. In chapter 3 methodology for constructing the model will be described. Chapter 4 reflects the results of the research experiment will be described. Chapter 5 will contain the social and environmental impact, ethical aspects, and sustainability plan of the project and Chapter 6 contains the summary, conclusion, future research plan.

CHAPTER 2

Background

2.1 Terminology

Here are some definitions of terms used in this article, to prevent misunderstanding due to ambiguity:

Machine learning [28] is a collection of techniques that allow computers to learn from data to make predictions and develop them.

Transfer learning [29] is a form of machine learning where a model learned on one assignment is re-purposed on a second similar assignment.

A Dataset is a table with the knowledge that the program learns from. The dataset includes the characteristics and the target to be projected. The dataset is called training data as used for the induction of a model.

An ophthalmologist [30] is a specialist in eye and vision treatment who is a medical or osteopathic practitioner.

Retinal fundus images are oriented on accurate orientation on the macula, clear visualization of the parafoveal arteries and more than two diameters of the disc around the fovea, and accurate measurement of the optic disc and the vascular arcades.

2.2 Related Works

Numerous researchers have worked in detecting Diabetic Retinopathy. In this section, some of the researches contributions in this field are described. Several machine learning and deep learning methods have been proposed in recent years for detecting Diabetic Retinopathy.

Zeynab et al. [1] proposed a Random transformed and Visibility graph method. They used contrast limited adaptive histogram equalization (CLAHE). The 1D image was generated

from a 2D image using Random transformed. Then the visibility graph was generated and DR grading was classified using ECOC. They generate an accuracy of 97.92%.

Romany [2] proposed a deep learning-based model for DR detection. They proposed AlexNet DNN based model with LDA based feature selection. They generate an accuracy of 97.93%.

ElTanboly et al. [4] proposed an automated approach in detecting diabetic retinopathy from SD-OCT images. They perform their work dividing it into three steps: segmenting the 12 layers of the retina, extract features (reflectivity, tortuosity, foveal angle, thickness) from OCT-images, and determined the difference between these layers by statistical analysis.

Pratt et al. [6] proposed a convolutional neural network and data augmentation approach to detect diabetic retinopathy. They trained their model on a dataset of 80000 images and generates an accuracy of 75%.

Ishtiyaque Ahmad [21] proposed a feature extraction technique. He detected five features namely optic disc, blood vessels, exudate, microaneurysm, and hemorrhage from the fundus image. He later applied SVM and ANN to detect Diabetic retinopathy. SVM generates an accuracy of 54.36% and ANN generates an accuracy of 59.22% on 61 test images.

M.R.K. Mookiah, U. Rajendra Acharya et al. [26] proposed a hybrid feature extraction techniques to distinguish two stages of DR from normal. They have implemented Probabilistic Neural Network, Decision tree, and SVM classifier to detect diabetic retinopathy. They have achieved an average accuracy of 96.15%, sensitivity of 96.27%, and specificity of 96.08%.

Balint Antal, Andras Hajdu [27] proposed a feature extraction technique followed by the implementation of different machine learning algorithms (Alternative Decision Tree, kNN, AdaBoost, Multilayer Perception, Naïve Bayes, Random Forest, SVM, Pattern classifier) to classify between a diseased and a normal retina. They have achieved an accuracy of 90% with a sensitivity of 90%, and specificity of 91%.

2.3 Comparative Analysis and Summary

A number of methods have been applied by different researchers to detect diabetic retinopathy. Some researchers have attained an accuracy higher than 95% which is higher than mine, but they have used a much-developed system and a large number of data to train their models. Where the accuracy of this proposed model is 94%. This is quite satisfactory compared to most other works done by different researchers. Moreover, many works have been done on the binary classification of diabetic retinopathy. In this project, I identified five stages of diabetic retinopathy.

Since finding a difference in fundus image is a complicated task. So picking up one algorithm that best fit for prediction purposes is a difficult task. Therefore, a number of algorithms were applied to find the best method.

- Sequential Convolutional Neural Network
- Deep learning architecture with pre-trained weight: DenseNet-121, DenseNet-201, VGG19, Inception V3, and XceptionNet.

Different CNN architectures have generated different accuracy. CNN generated the least accuracy, all other models were trained initially on a pre-trained using ImageNet. All the transfer learning algorithms generated accuracy ranging from 91-94%. The highest accuracy of 94% was generated by DenseNet121.

2.4 Scope of the Problem

1. People can find the stage of disease inputting the fundus image of the retina. It cannot detect Diabetic Retinopathy by other features.
2. Fundus images can only be obtained by funduscopy which is not available for general people. Therefore, people from all walks of life cannot test their eyes.
3. The retinal image shows significant variation based on the geographical location of the people. It is very difficult to collect a significant number of images from all walks of people around the world to train the machine.
4. This proposed system can only identify diabetic retinopathy. But there are many other retinal diseases. But this application will not be able to identify them.

2.5 Challenges

1. To install necessary libraries
2. Preprocessing the image and to generate NumPy data
3. Selecting suitable algorithms
4. Classifications methods took a long time on my system.

CHAPTER 3

Research Methodology

3.1 Research Subject and Instrumentation

This project aims to identify five stages of Diabetic retinopathy. Detection of 5 stages of Diabetic Retinopathy is extremely difficult as symptoms do not appear on the retina before it becomes extremely serious. In this project, by trying some Deep Neural Network (DNN) I tried to create a pattern. I tried sequential Convolutional Neural Network (CNN), some CNN architecture like DenseNet-121, DenseNet-201, VGG19, Inception V3, XceptionNet. After implementing each of these algorithms I try to compare the output to find the best model.

For the development of the model, I have used Python 3.7 as the programming language and many libraries need to be installed to have different features of python. I have used Anaconda Navigator for quick access to the majority of the features. Jupyter Notebook application which is a built-in feature of Anaconda Navigator was used to run the code. To make the project usable by the common people an application needed to be developed that provides a portal where people can check the condition of their retinal. Flask web framework was used to deploy the developed model and for the implementation of Flask, I have used the Spyder IDE application. People can use different web surfing tools like Google Chrome, Microsoft Edge, etc. to use the web application.

3.2 Data Collection Procedure

To train the model, I had to provide a good amount of image data into the model. For this purpose, I downloaded a total of 3662 images from kaggle.com. [18] The image data was provided by Indian Aravind Eye Hospital who collected fundus images from different areas of India and reviewed by highly trained doctors. Dataset was provided with a CSV file containing the image ID and corresponding labeling. Among the total number of

dataset 1805 was labeled as no Diabetic Retinopathy, 370 as mild-DR, 999 as moderate-DR, 193 as Severe-DR, and 295 as Proliferative-DR.

3.3 Statistical Analysis

A total of 3662 images contain image of five different classes of image. Moreover, for training and validation purpose total dataset was divided into training and validation dataset. Table 3.3.1 illustrates the number of images belonging to each class, also number of images belonging to training dataset and number of images belonging validation dataset.

Table 3.3.1: Statistical analysis of images of each class of Diabetic Retinopathy

Class Name	Class Label	Total Data	Train Data	Test Data
No-DR	0	1805	1542	263
Mild-DR	1	370	320	50
Moderate-DR	2	999	833	166
Severe-DR	3	193	170	23
Proliferative-DR	4	295	247	48

3.4 Proposed Methodology

The whole process is divided into a number of sub-processes to successfully develop the model. Figure 3.1.1 illustrates the entire process that I have adopted in developing this project.

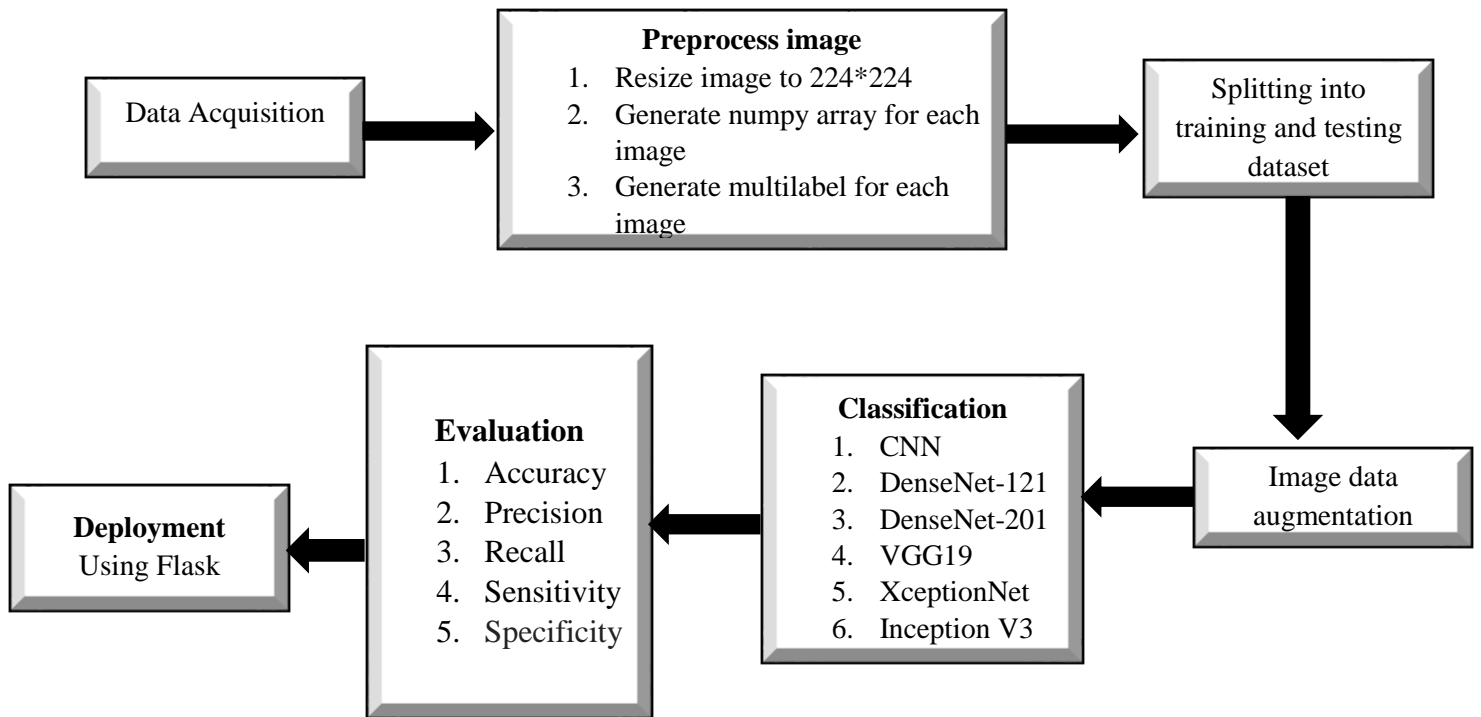


Figure 3.4.1: Block diagram of methodology

Preprocess Image:

The dataset contains images of different sizes. To make an effective model all the images should have a unique shape. For that purpose, I have resized all the images into 224*224 pixels. To feed the image into the model, I convert the image data into a NumPy array. Then I have done multi-labeling for each class. That is if an image belongs to stage-4 it has all the preceding stages because, retinopathy has to pass the mild, moderate, severe stage to be proliferative.

Splitting data:

For the training and testing purposes, I have used 85% of the total dataset as the training data and 15% of the total dataset as the testing purpose for the DenseNet architecture. And for other models, I have used 80% data as the training data and 20% as the testing data.

Image Data Augmentation:

Image data augmentation is a method that can be used to dynamically increase the size of a training dataset by generating updated model versions in the dataset. This is achieved by adding domain-specific methods to instances from training data generating new and

distinct teaching examples. I have used ImageDataGenerator [7] a built-in library of Keras for image augmentation. In this project I set the zoom range to 15% and enabled the horizontal and vertical flip true, so that the images were zoomed to 15% making the features more visible to the machine and enabling flipping allows the image to rotate on different axis according to need to generate the best outcome.

Choice of models:

Picking up a model to generate the best performance is a difficult task. One algorithm may generate better performance than the others. Again, accuracy depends on many other factors. So, in order to attain the best model, it is wise to try a number of algorithms to have the highest accuracy.

I will implement a Convolutional Neural Network (CNN). I will also be using some transfer learning approach using XceptionNet, DenseNet-121, DenseNet-201, Inception V3, VGG 19 with predefined weight on ImageNet. Then I will evaluate the performance of each model to find the best model for my project.

Convolutional Neural Network (CNN) [8]:

All the input images were resized to 224*224 pixels. Image pixels were then converted to unsigned integer type NumPy data. I have also done a multi-labeling on the dataset which decreases the possibility of misclassification to a greater extent. For training and testing purposes, the dataset was then divided into training and validation sets where the training set consist of 80% of the total images, and the validation set consists of the remaining 20% with a predefined random state to 2020. A sequential model of CNN was created. Relu was the activation layer for the hidden layer. The output layer has 5 output layers with the activation function set to sigmoid. A callback function was created using the Keras callback library to save the best model during the training phase. The model was trained using 15 epochs and generated an accuracy of 88.92%. Figure 3.1.2 shows a basic overview of CNN architecture.

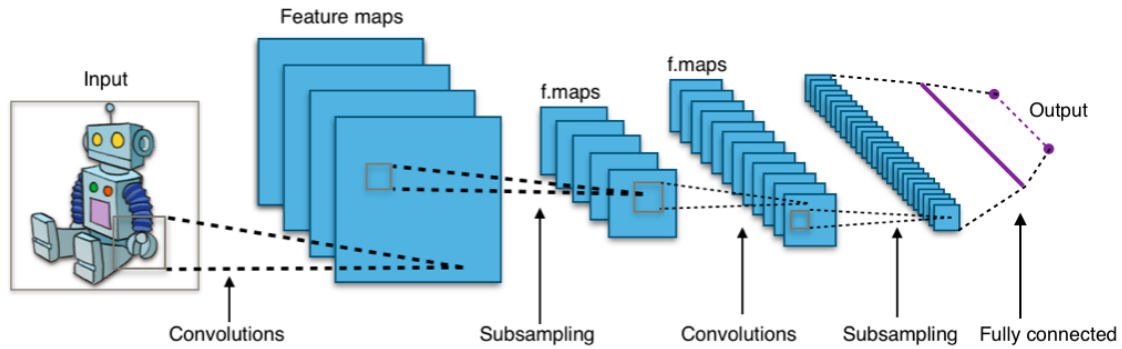


Fig 3.4.2: Typical CNN architecture [8]

Transfer learning models:

- XceptionNet:** XceptionNet [9][10] is a 126 layers deep convolutional neural network. We can perform training on our images using XceptionNet using a pre-trained weight of ImageNet, which will guide the model for better performance.[9] Keras's application library provides the XceptionNet function. Images were augmented first before feeding it into the model with a pre-defined batch size of 32. After initializing the model I add the following layers:
 - 2x convolution layer of 64 channel of 3x3 KERNEL and use same padding
 - 1x maxpool layer of 2x2 pool size and stride 2x2
 - 2x convolution layer of 128 channel of 3x3 KERNEL and use same padding
 - 1x maxpool layer of 2x2 pool size and stride 2x2
 - 3x convolution layer of 256 channel of 3x3 KERNEL and use same padding
 - 1x maxpool layer of 2x2 pool size and stride 2x2
 - 3x convolution layer of 512 channel of 3x3 KERNEL and use same padding
 - 1x maxpool layer of 2x2 pool size and stride 2x2
 - 3x convolution layer of 512 channel of 3x3 KERNEL and use same padding
 - 1x maxpool layer of 2x2 pool size and stride 2x2

After that I add

- 1x Dropout layer
- 1x Dense Sigmoid layer of 5 units

After setting the model I compiled the model. For that, I used Adam optimizer with setting the learning rate to 0.00005. To check and save the best model I have used Keras built-in callback function. For that, I imported ModelCheckPoint and EarlyStopping methods from Keras. Once the model finishes its training phase I evaluated the performance of the model. XceptionNet generated an accuracy of 92.25%. Figure 3.1.3 illustrates the data through the layers of XceptionNet.

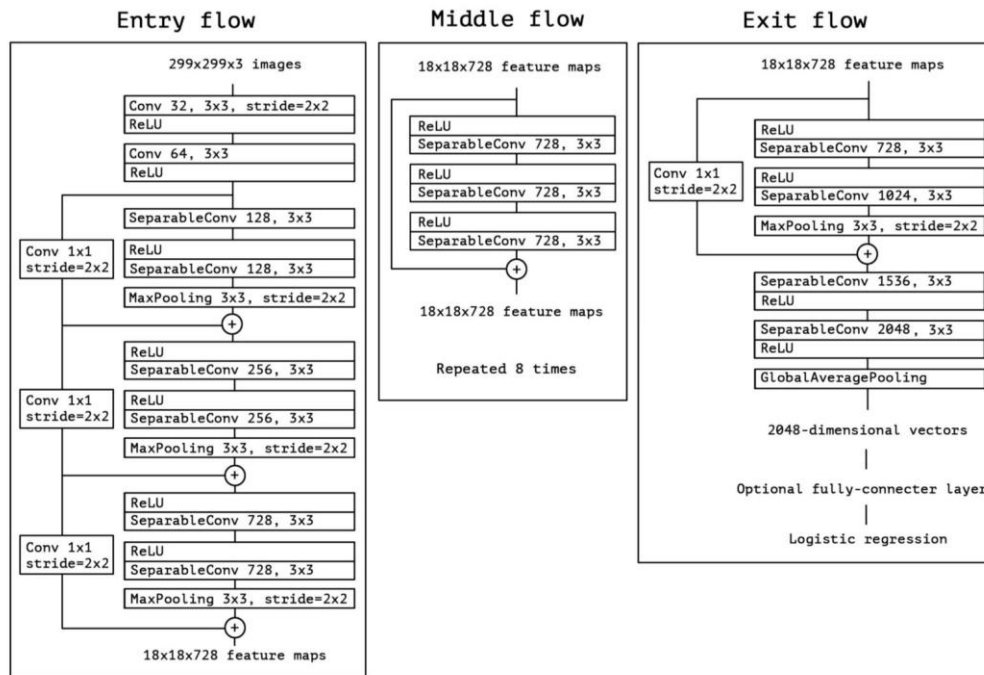


Fig 3.4.3: XceptionNet Architecture [10]

- Inception V3:** Inceptionv3 [11][12] is a convolutional neural network that supports image recognition and object detection and has begun as a Googlenet module. Just like ImageNet can be thought of as a database of classified visual objects, Inception aims to identify objects in the computer vision world.[11] Inception V3 is 159 layers deep.[12] For training my model using inception v3 I have used pre-trained weight on ImageNet and for test purpose, I kept 20% of my dataset. After initializing the model I add the following layers:
 - 2x convolution layer of 64 channel of 3x3 KERNEL and use same padding
 - 1x maxpool layer of 2x2 pool size and stride 2x2
 - 2x convolution layer of 128 channel of 3x3 KERNEL and use same

padding

- 1x maxpool layer of 2x2 pool size and stride 2x2
- 3x convolution layer of 256 channel of 3x3 KERNEL and use same padding
- 1x maxpool layer of 2x2 pool size and stride 2x2
- 3x convolution layer of 512 channel of 3x3 KERNEL and use same padding
- 1x maxpool layer of 2x2 pool size and stride 2x2
- 3x convolution layer of 512 channel of 3x3 KERNEL and use same padding
- 1x maxpool layer of 2x2 pool size and stride 2x2

I add ReLU as the activation function in each layer. I have taken Adam optimizer and sigmoid as the activation function and the learning rate was set to 0.00005. To check and save the best model I have used Keras built-in callback function. For that, I imported ModelCheckpoint and EarlyStopping methods from Keras. Once the model finishes its training phase I evaluated the performance of the model. The accuracy obtained using Inception V3 is 92.14%. Figure 3.1.4 shows the architecture of Inception v3 model.

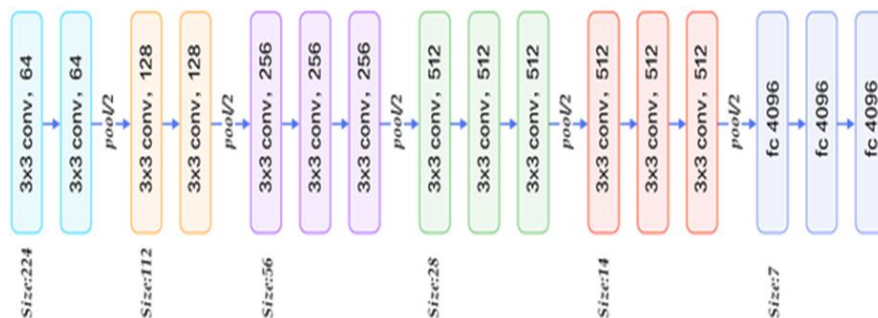


Fig 3.4.4: Inception V3 architecture [12]

- **VGG19:** VGG19 [13] is a convolutional neural network with 26 layers depth. To implement the architecture I imported the model from Keras library. I have also imported other necessary libraries like pandas, train-test-split, ImageDataGenerator from Keras. Image data was preprocessed to generate a NumPy array that I used for the training purpose. Then a sequential model was created and VGG 19 was

imported. Image data augmentation was done using ImageDataGenerator. After initializing the model I add the following layers:

- 2x convolutional layer of 64 channel of 3x3 KERNEL and use same padding.
- 1x maxpool layer of 2x2 pool size and stride 2x2
- 2x convolutional layer of 128 channel of 3x3 KERNEL and use same padding
- 1x maxpool layer of 2x2 pool size and stride 2x2
- 3x convolutional layer of 512 channel of 3x3 KERNEL and use same padding
- 1x maxpool layer of 2x2 pool size and stride 2x2
- 3x convolutional layer of 512 channel of 3x3 KERNEL and use same padding
- 1x maxpool layer of 2x2 pool size and stride 2x2
- 3x convolutional layer of 512 channel of 3x3 KERNEL and use same padding
- 1x maxpool layer of 2x2 pool size and stride 2x2

I used a 2D max-pooling layer and a dropout layer. Rectified Linear Unit (ReLU) was used in the hidden layer as the activation function and sigmoid in the output layer. 80% of the total dataset was used for training purposes and 20% was used for validation purposes. I set a callback function to save the best model. Based on 15 epochs the model generates an accuracy of 93.45% with a sensitivity of 64.76% and a specificity of 95.74%. fig 3.2.4 describes VGG19 architecture.

→ 3x convolutional layer of 512 channel of 3x3 KERNEL and use same padding

→ 1x maxpool layer of 2x2 pool size and stride 2x2

Rectified Linear Unit (ReLU) was used as the activation function. This prevents the passing of negative values to the next layer. Once the model is prepared I compiled it. I used Adam optimizer to reach global minima. The learning rate was set to 0.00005. To check and save the best model I have used Keras built-in callback function. For that, I imported ModelCheckpoint and EarlyStopping methods from Keras. Once the model finishes its training phase I evaluated the performance of the model. The model generates an accuracy of 93% with a specificity of 96% and a sensitivity of 68%.

- **DenseNet-121: [15][16]** DenseNet-121 is a convolutional neural network with 121 layers deep. Like 201 here each layer obtains additional inputs from all preceding layers, concat them and it also transmits its feature to all subsequent layers. DenseNet requires fewer parameters than a traditional CNN. CNN can be more accurate and efficient if the layers contain shorter connections between the input and output layers. When in a normal CNN K number of nodes has a K number of connections, DenseNet has $K(k+1)/2$ number of connections. DenseNet uses a convolutional layer and a maximum pooling layer. Then DenseNet uses four dense blocks and transition layers. Dense blocks and Transition layers are two main components of DenseNet. A dense block has multiple conv_block units. Where the number of output channels is the same. A transition layer controls the complexity of the model. It reduces the number of channels by using a 1*1 convolutional layer and halves the height and weight of the average pooling layer with a stride of 2, further reducing the complexity of the model. For training with DenseNet-121, I have used 15% of the total dataset as the validation set and remaining as the training set. The model generates an accuracy of 94% with a sensitivity of 69% and a specificity of 96%. Figure 3.1.6 shows the architecture of DenseNet.

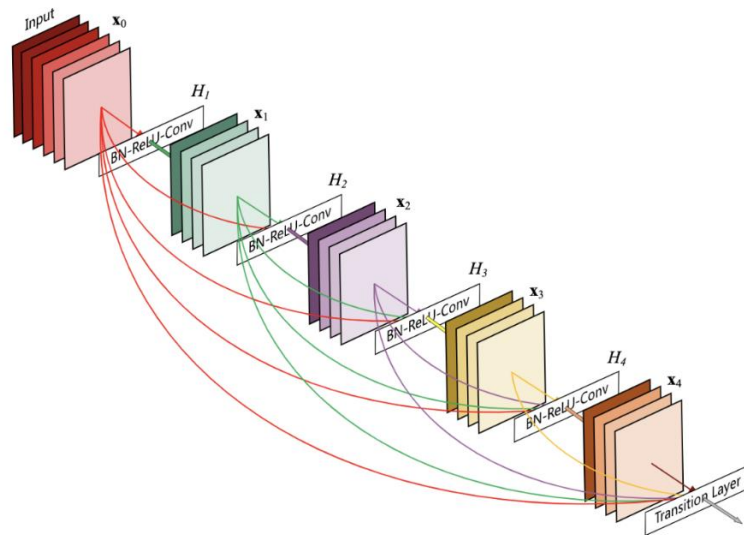


Fig 3.4.6: DenseNet architecture [17]

Application Develop:

I have developed a web application using Flask that will recognize the stage of diabetic retinopathy simply by uploading the retinal fundus image. DenseNet-121 is implemented for the prediction purpose in the application. Figure 3.1.7 illustrates the webpage a user will see whenever he will visit the website. Figure 3.1.8 illustrates the output of the system whenever user selects an image from his system.

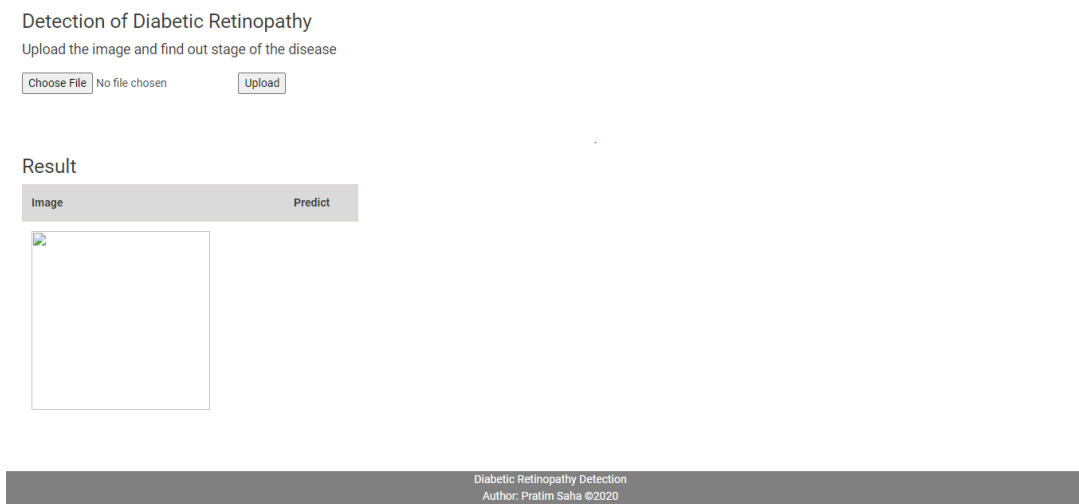


Fig 3.4.7: Page to upload image

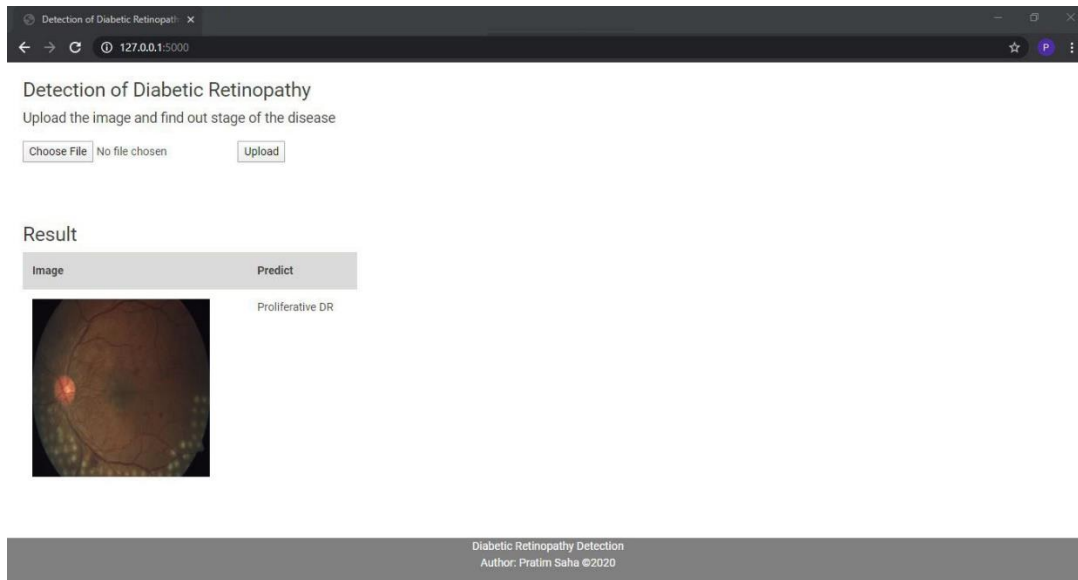


Fig 3.4.8: web page with predicted diabetic retinopathy

3.5 Implementation Requirements

Here is a list of all the tools, devices, languages, and programs that I have used to develop and deploy my project.

- **Back end:**
 1. Python (3.7.4) [31]
 2. CPU: Intel core i3[32]
 3. RAM: 4GB [33]
 4. Operating system: Windows 10 64bit [34]
 5. Keras [35]
 6. PIL [36]
 7. Numpy [37]
 8. Tensorflow [38]
 9. Sklearn [39]
 10. Pandas [40]
 11. Flask [41]

Python is a high level programming language that is object oriented and has dynamic semantics. Python supports functional and objects oriented programming language. It has many libraries like Numpy that is used for working with arrays. Numpy stands for Numerical Python. PIL stands for Python Imaging Library which is a free and open-source additional library used for opening, manipulating, and saving image file formats. Scikit-learn also is known as sklearn is a machine learning library for python. It provides many classifications, regression, and clustering algorithms. It also incorporates NumPy and SciPy. In this project, I have used sklearn for splitting training and testing data using the `train_test_split()` function and also used it for generating an accuracy score. Keras is a neural network library running on top of TensorFlow. It is a high-level API for the human being. Pandas is a python package used for fast, flexible, and expressive data analysis and manipulation. I have used it to import CSV files and handling of data frames. To develop the web application I have used Flask which is a micro web framework of python.

- **Front end:**

1. [HTML \[42\]](#)
2. [CSS \[42\]](#)
3. [JavaScript \[42\]](#)
4. [Bootstrap \[42\]](#)

HTML is a markup language that is used to build the skeleton of a webpage. Cascading style sheet or CSS is a stylesheet language used to determine how elements are organized on-screen or on media. Bootstrap is a framework that is used to use the feature of CSS efficiently. JavaScript is an object-oriented language used to make a webpage responsive.

CHAPTER 4

Experimental Results and Discussion

4.1 Experimental Setup

In this chapter, the experiment of the project is described. Different models were implemented to find the best model suited for detecting diabetic retinopathy. All the models were trained on 85% of the total dataset as the training set and the remaining 15% as the validation set. After implementing each model, their accuracy and loss were evaluated. The confusion matrix was plotted for each of the models for visualizing the performance of the model. Following formulas are used to find out the accuracy, sensitivity and specificity [43].

$$\text{accuracy} = \frac{\text{true positive} + \text{true negative}}{\text{total}}$$

$$\text{Sensitivity} = \frac{\text{true positive}}{\text{true positive} + \text{false negative}}$$

$$\text{Specificity} = \frac{\text{true negative}}{\text{true negative} + \text{false positive}}$$

These formulas generate class-wise evaluation of accuracy, sensitivity, and specificity. To find the average accuracy, sensitivity, and specificity of the model I averaged the value of each individual class. So finding the average accuracy, sensitivity and specificity I used the following formula:

$$\text{average accuracy} = \frac{\text{sum}(\text{accuracy})}{\text{len}(\text{accuracy})}$$

$$\text{average sensitivity} = \frac{\text{sum}(\text{sensitivity})}{\text{len}(\text{sensitivity})}$$

$$\text{average specificity} = \frac{\text{sum}(\text{specificity})}{\text{len}(\text{specificity})}$$

Finally, the model with the highest accuracy was chosen as the desired model.

4.2 Experimental Results & Analysis

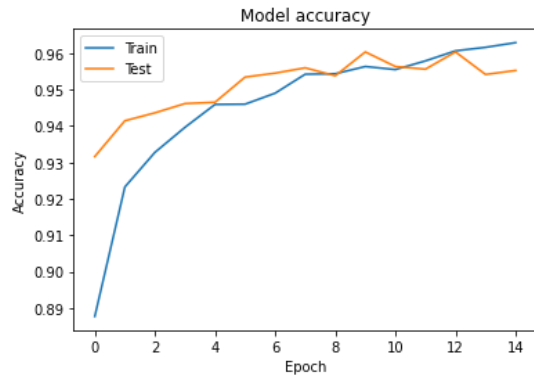
- Architecture of CNN. Table 4.2.1 illustrates a different convolutional neural network architecture that I have used in this project with corresponding depth, training accuracy, training loss, testing accuracy, testing loss the models experience during the training phase.

Table 4.2.1: Experimental Result of different neural network model

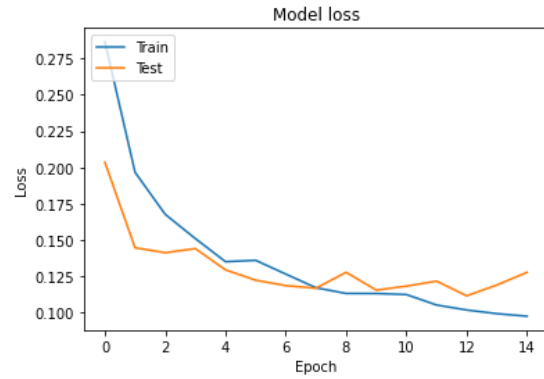
Model	Parameters	Depth	Train Accuracy	Train Loss	Test Accuracy	Test loss
DenseNet-121	8,062,504	121	0.9729	0.0691	0.9622	0.1035
DenseNet-201	20,242,984	201	0.9600	0.1045	0.9589	0.1083
XceptionNet	22,910,480	126	0.9712	0.0751	0.9604	0.1027
VGG19	143,667,240	26	0.9607	0.1014	0.9604	0.1115
Inception V3	23,851,784	159	0.9607	0.0990	0.9527	0.1269
CNN	995,685	7	0.9015	0.2559	0.9102	0.2302

VGG19

Graph: figure 4.2.1 (a) shows the training and testing accuracy of the model and (b) shows the training and testing loss of the model.



(a)

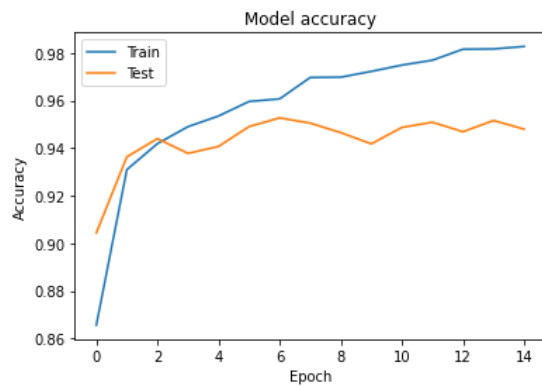


(b)

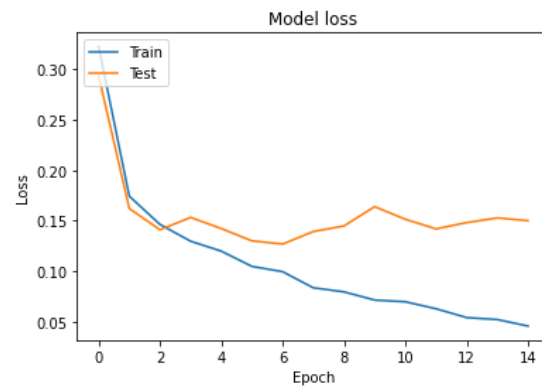
Figure: 4.2.1: (a) VGG19 model accuracy; (b) VGG19 model loss

InceptionV3

Graph: figure 4.2.2 (a) shows the training and testing accuracy of the model and (b) shows the training and testing loss of the model.



(a)



(b)

Figure 4.2.2: (a) Inception model accuracy; (b) Inception model loss

Xception

Graph: figure 4.2.3 (a) shows the training and testing accuracy of the model and (b) shows the training and testing loss of the model.

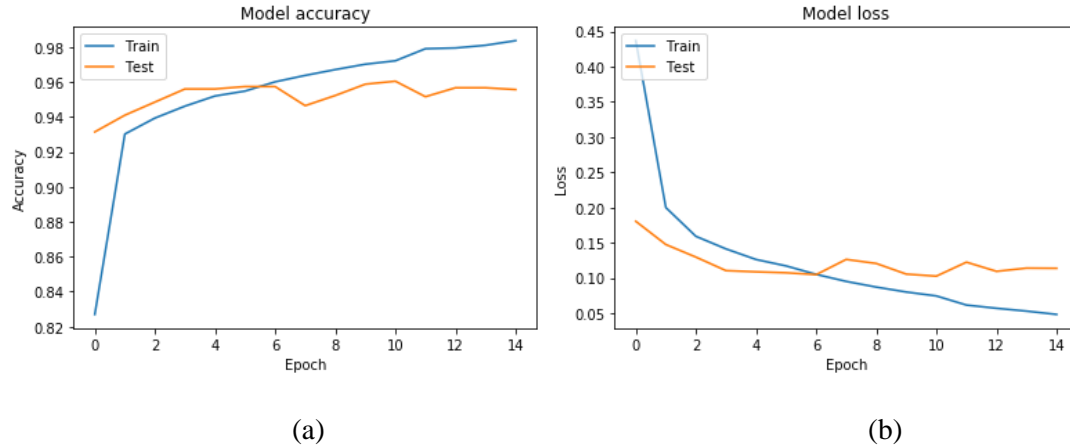


Figure 4.2.3: (a) XceptionNet model accuracy; (b) XceptionNet model loss

DenseNet-201

Graph: figure 4.2.4 (a) shows the training and testing accuracy of the model and (b) shows the training and testing loss of the model.

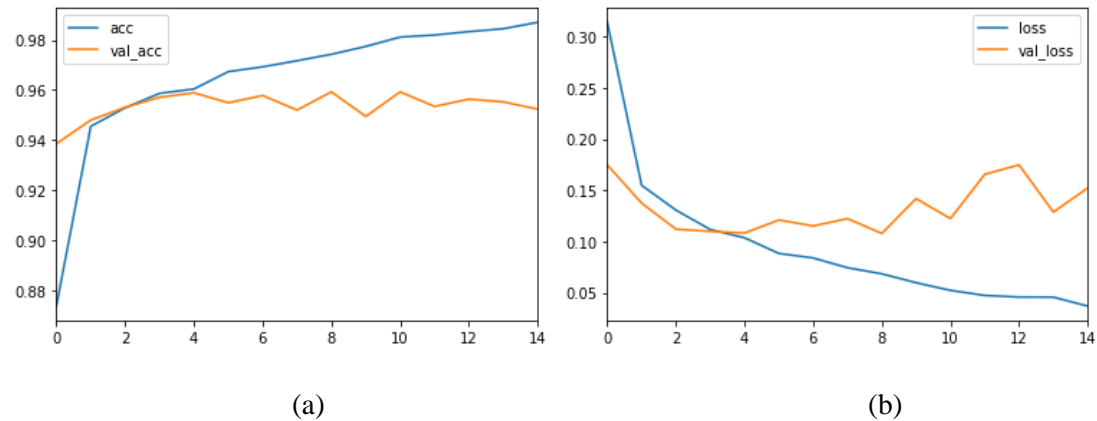


Figure 4.2.4: (a) DenseNet-201 model accuracy; (b) DenseNet-201 model loss

DenseNet-121

Graph: figure 4.2.5 (a) shows the training and testing accuracy of the model and (b) shows the training and testing loss of the model.

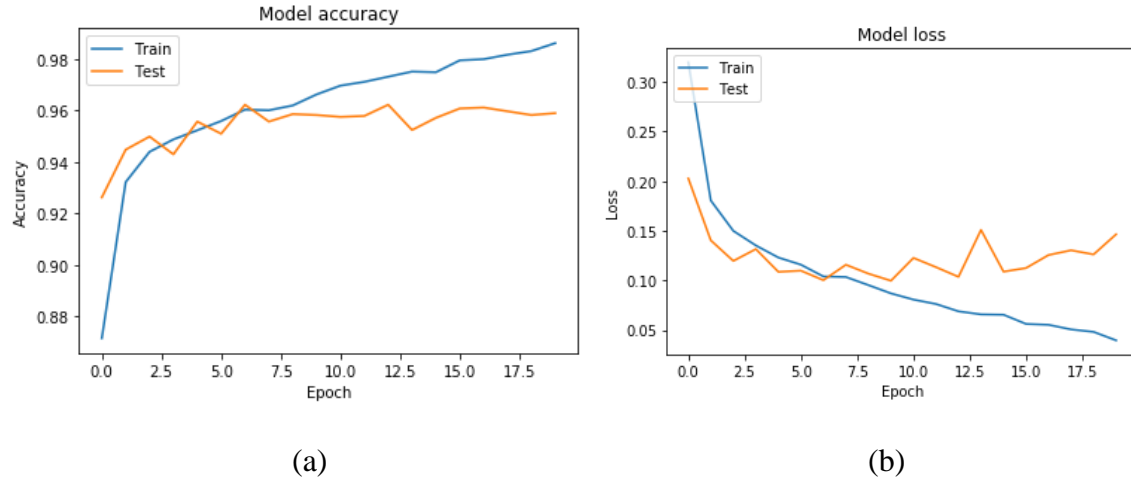


Figure 4.2.5: (a) DenseNet-121 model accuracy; (b) DenseNet-121 model loss

I have plotted a confusion matrix to determine the accuracy of CNN and other architectures. For Xception accuracy is 92.25%, for VGG19 93.45%, for Inception 92.14%, for DenseNet-201 93.09%, for DenseNet-121 93.82% and for CNN 87.85%. Figure 4.2.6 shows a bar chart diagram that describes the accuracy of different models that I could achieve for our given dataset and used CNN architecture.

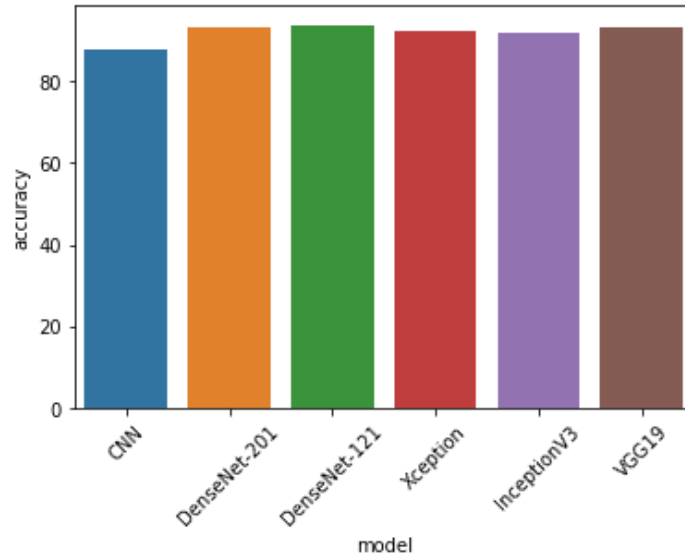


Figure 4.2.6: Accuracy of different models

For running each model I set the same number of steps per epoch which is 97. I set the number of epochs 15 and 20 for these models but in all cases, the highest accuracy was obtained in less than 10 epochs except for the CNN architecture where the highest accuracy was obtained in the 14th epoch. I set validation kappa in each of this architecture which checks the validation loss for each epoch and saves the best weight of the model containing the least validation loss.

4.3 Discussion

After comparing the accuracy of different models we can conclude that DenseNet-121 generates the best accuracy. So my proposed model is DenseNet-121 for detecting diabetic retinopathy from the retinal fundus image. Table 4.4.1 shows the class-wise performance of DenseNet-121 architecture.

Table 4.3.1: Class-wise performance of DenseNet-121

	Stage-0	Stage-1	Stage-2	Stage-3	Stage-4
Accuracy	98.91%	92.36%	85.27%	92.26%	94.72%
Sensitivity	98.65%	37.25%	77.94%	46.15%	50%
Specificity	99.20%	97.99%	87.68%	94.66%	98.23%

CHAPTER 5

Impact on Society, Environment and Sustainability

5.1 Impact on Society

This project is going to have a great impact on society. People from all walks of life can use the application implemented through this project to check if they have any retinopathy, they can also learn the stage their retinopathy is. This project is a blessing for the rural poor people for whom it is not always possible to consult an ophthalmologist due to the lack of money and the number of doctors available. Moreover, people can check the disease in a few seconds which usually takes a good time to check manually. So this project has a positive impact on society.

5.2 Impact on Environment

Any work has a more or less environmental impact. This project has a positive impact on the environment. There is a possibility of emitting any toxic substances or anything that harms the environment. Moreover, this project is going to help people to save their eyesight by early detection of their disease. So from that point of view, it is going to help the environment. So, this project has a positive impact on the environment.

5.3 Ethical Aspects

Some ethical issues arise while implementing a machine learning algorithm for medical diagnosis. Firstly the authority of using the dataset. Many end-user license agreements restrict the use of the dataset. Using such a dataset without permission raises ethical issues. For this project, I have collected datasets from open-source data repository Kaggle.com. Datasets of kaggle are open to any researchers.

Another issue arises with the fact that a doctor explains why and how disease occurs and the possible outcome of that disease. But how a machine detects that disease is completely unknown to the users. So, it might raise a question about the correctness of the diagnosis.

A possible solution to the problem is releasing the program under open-source licenses. The source code of this project is open to any person who is interested to have a deeper look into the project implementation.

5.4 Sustainability Plan

This project aims to lessen the sufferings of the common people. To ensure its sustainability more advancement will be brought to the project. Implementation of this project will no longer be confined to web applications only, rather an android version, as well as an iOS version software, will be implemented. The existing web page will be updated to attract more people and there will be a proper guideline to the users so that they know how to use it. More algorithms will be applied in the future in an attempt to increase the performance of the model. I have a plan to not confine the project in detecting diabetic retinopathy only. Rather, it will test other eye diseases at the same time to ensure a better life of the people.

CHAPTER 6

Summary, Conclusion, Recommendation and Implication for Future Research

6.1 Summary of the Study

Diabetic retinopathy is one of the most leading causes of blindness in the world. Early detection of diabetic retinopathy is a must. An automated approach can save the eyesight of a lot of people around the globe. In this project, I have used a 3662 retinal fundus image for the detection of four stages of diabetic retinopathy and also the healthy eye. For selecting the best machine learning model for this purpose, I have implemented some architectures. By comparing the performance of these models I have chosen DenseNet-121 as this generates the best accuracy among the architecture I have implemented in this project.

6.2 Conclusions

In this project, I have used the dataset collected from different areas of India, so this project has created an opportunity for the people of the Indian sub-continent to have to get a more correct prediction of their retinopathy detection since most works have been done on Caucasian peoples' fundus image. Although the proposed DenseNet-121 model has generated an accuracy of 94% with a specificity of 96%, the sensitivity is as low as 69%. This is due to the high variation of the dataset and the limited number of images per class. Sensitivity can be increased by providing a more labeled dataset per class. In the future, a more labeled dataset will be used along with the application of other deep convolutional neural network model in an attempt to increase the performance in detecting diabetic retinopathy of different stages.

6.3 Implication for Future Study

In this work, I have focused on detecting 5 stages of diabetic retinopathy by implementing different deep CNN architectures. In the future, I would focus on detecting different retinal diseases like Retinal tear, Retinal detachment, macular hole, macular degeneration, Retinitis using retinal fundus image. I will try to build a model that will detect all these diseases from a single image along with the stage of the disease. I will also try other deep learning models like NasNetMobile, NiN, FractalNet, etc.

REFERENCES

- [1] Z. Mohammadpoury et al., “Automatic identification of diabetic retinopathy stages by using Fundus images and Visibility Graph Method”, Measurement, Volume 140, Pages 133-141, July 2019
- [2] Romany F. Mansour, “Deep-learning-based automatic computer-aided diagnosis system for diabetic retinopathy”, Korean Society of Medical and Biological Engineering and Springer-Verlag GmbH Germany 2017, August 2017
- [3] Rajib Raman et al., “Fundus photograph-based deep learning algorithms in detecting diabetic retinopathy”, The Royal College of Ophthalmologists, Eye (2019) 33:97–109, November 2018
- [4] Ahmed H. ElTanboly, Agustina Palacio et al. “An automated approach for early detection of diabetic retinopathy using SD-OCT image”, Frontiers in Bioscience, Elite 10, Pages 197-207, January 2018
- [5] Muhammad Sharif, Jamal Hussain Shah, “Automatic Screening of Retinal Lesion for Gradin Diabetic Retinopathy”, The International Arab Journal of Information Technology, Vol. 16, July 2019
- [6] Harry Pratt et al., “Convolutional Neural Networks for Diabetic Retinopathy”, Procedia Computer Science, Volume 90, Pages 200-205, July 2016
- [7] tf.keras.preprocessing.image.ImageDataGenerator available at https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator last accessed on October 3, 2020 at 1:45AM
- [8] Convolutional Neural Network available at https://en.wikipedia.org/wiki/Convolutional_neural_network last accessed on September 27, 2020 at 1:54 AM
- [9] Xception convolutional neural network available at <https://www.mathworks.com/help/deeplearning/ref/xception.html> last accessed on September 27, 2020 at 1:55 AM
- [10] Xception|Kaggle available at <https://www.kaggle.com/keras/xception> last accessed on September 27, 2020 at 2:02 AM
- [11] Inceptionv3 available at <https://en.wikipedia.org/wiki/Inceptionv3> last accessed on September 27, 2020 at 2:04 AM
- [12] Keras applications available at <https://keras.io/api/applications/> last accessed on September 27, 2020 at 2.10 AM

- [13] Architecture of the FCN-VGG19 available at <https://www.researchgate.net/figure/Architecture-of-the-FCN-VGG19-adapted-from-Long-et-al-2015-which-learns-to-combine_fig1_331258180> last accessed on September 27, 2020 at 2:15 AM
- [14] DenseNet201 available at <<https://www.mathworks.com/help/deeplearning/ref/densenet201.html%20https://www.mathworks.com/help/deeplearning/ref/densenet201.html>> last accessed on September 27, 2020 at 2:20 AM
- [15] Review: DenseNet — **Dense Convolutional Network** (Image Classification) available at <<<https://towardsdatascience.com/review-densenet-image-classification-b6631a8ef803>>> last accessed on September 27, 2020 at 2.20 AM
- [16] Understanding and Visualizing DenseNets available at <<https://towardsdatascience.com/understanding-and-visualizing-densenets-7f688092391a>> last accessed on September 27, 2020 at 2.26 AM
- [17] CNN ARCHITECTURES: DENSENET available at <<https://machinelearningtokyo.com/2020/04/21/cnn-architectures-densenet/>> last accessed on September 27,2020 at 2.40 AM
- [18] APTOS 2019 Blindness Detection available at <<https://www.kaggle.com/c/aptos2019-blindness-detection/data>> last accessed on September 27, 2020 at 2.45 AM
- [19] Zvia Burgansky-Eliasb et al., “Optical Coherence Tomography Machine Learning Classifier for Glaucoma Detection: A Preliminary Study”, IOVS, Vol 46, No.11, , November 2005
- [20] Varun Gulshan et al., “Development and Validation of a Deep Learning Algorithm in Retinal Fundus Photographs”, American Medical Association, November 29,2016
- [21] Ishtiyaque Ahmad, “Automatic Detection of Diabetic Retinopathy from Fundus Images Using Image Processing and Artificial Neural Network”, BUET, January 2019
- [22] Diabetic Retinopathy-Wikipedia available at <https://en.wikipedia.org/wiki/Diabetic_retinopathy> last accessed on September 27, 2020 9AM.
- [23] The worldwide epidemic of diabetic retinopathy available at <<[34](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3491270/#:~:text=Globally%2C%20the%20number%20of%20people,prompt%20action%20is%20not%20taken.>> last accessed on September 27, 2020 at 10 AM.</p>
<p>[24] Yingfeng Zheng, Mingguang He, and Nathan Congdon, “The worldwide epidemic of diabetic retinopathy. Indian journal of ophthalmology”, 60(5):428, 2012.</p>
<p>[25] Nathan Silberman, Kristy Ahrlich, Rob Fergus, and Lakshminarayanan Subramanian, “Case for automated detection of diabetic retinopathy.” In Proceedings of the AAAI Spring Symposium: Artificial Intelligence for Development, California, USA, March 22-24, 2010.</p>
<p>[26] M.R.K Mookiah, U. Rajendra Acharya et al., “Evolutionary algorithm based classifier parameter tuning for automatic diabetic retinopathy grading: A hybrid feature extraction approach”, Knowledge-based-system, Volume 39, Pages 9-22, February 2013.</p>
</div>
<div data-bbox=)

- [27] Balint Antal, Andras Hajdu, “An ensemble-based system for automatic screening of diabetic retinopathy”, Knowledge-based systems, Volume 60, Pages 20-27, April 2014
- [28] Terminology| Interpretable Machine learning available at
<<https://christophm.github.io/interpretable-ml-book/terminology.html>> last accessed on September 27, 2020 at 6:00 PM.
- [29] Transfer learning available at
<[https://en.wikipedia.org/wiki/Transfer_learning#:~:text=Transfer%20learning%20\(TL\)%20is%20a,when%20trying%20to%20recognize%20trucks.](https://en.wikipedia.org/wiki/Transfer_learning#:~:text=Transfer%20learning%20(TL)%20is%20a,when%20trying%20to%20recognize%20trucks.)> last accessed on September 27,2020 at 5:00PM.
- [30] What is an Ophthalmologist? Available at <<https://www.aaopt.org/eye-health/tips-prevention/what-is-ophthalmologist>> last accessed on September 27,2020 at 6:10PM.
- [31] What is python available at <<https://www.python.org/doc/essays/blurb/>> last accessed on September 7, 2020 at 6:15PM.
- [32] Intel Core i3 available at <<https://www.intel.com/content/www/us/en/products/processors/core/i3-processors.html>> last accessed on September 29, 2020 at 1PM.
- [33] Random-Access-Memory available at <https://en.wikipedia.org/wiki/Random-access_memory> last accessed on September 29, 2020 at 1 PM.
- [34] Windows 10 available at
<https://en.wikipedia.org/wiki/Windows_10#:~:text=Windows%2010%20is%20a%20series,public%20on%20July%2029%2C%202015.> last accessed on September 29, 2020 at 1:10 PM.
- [35] Keras application available at <<https://keras.io/api/applications/>> last accessed on September 30, 2020 at 1 PM
- [36] Pillow available at <<https://pillow.readthedocs.io/en/stable/>> last accessed on September 30, 2020 at 1 PM.
- [37] Numpy available at <<https://numpy.org/>> last accessed on October 2, 2020 at 5:50PM
- [38] Tensorflow available at <<https://www.tensorflow.org/>> last accessed on October 2, 2020 at 5PM
- [39] Scikit learn available at <<https://scikit-learn.org/stable/>> last accessed on October 2,2020 at 9PM
- [40] Pandas available at <<https://pandas.pydata.org/>> last accessed on October 1,2020 at 8 AM
- [41] Welcome to Flask available at <<https://flask.palletsprojects.com/en/1.1.x/>> last accessed on October 1, 2020 at 8:30 AM
- [42] W3school online web tutorial available at <<https://www.w3schools.com/>> last accessed on October 2,2020 at 4PM
- [43] Basic evaluation measures from confusion matrix available at
<[https://classeval.wordpress.com/introduction/basic-evaluation-measures/#:~:text=Sensitivity%20\(SN\)%20is%20calculated%20as,true%20positive%20rate%20\(TPR\).&text=Sensitivity%20is%20calculated%20as%20the,number%20of%20positives%20\(P\).](https://classeval.wordpress.com/introduction/basic-evaluation-measures/#:~:text=Sensitivity%20(SN)%20is%20calculated%20as,true%20positive%20rate%20(TPR).&text=Sensitivity%20is%20calculated%20as%20the,number%20of%20positives%20(P).)>
last accessed on October 2, 2020 at 6PM.

APPENDICES

Appendix A: Research Reflection

After this project, I have gained a huge knowledge of Machine learning and Diabetic retinopathy. After exploring the experimental result and overviewing the related work, I have learned that it is possible to identify diabetic retinopathy using a computer within seconds. In the future, if I get a chance to work on a similar project I will try to implement a machine learning approach for detecting other diseases to lessen the suffering of the people.

Appendix B: Related Issues

The collection of the dataset and preparing them for the project is one of the most important issues. I faced some difficulties in preprocessing the images, i.e. resizing the image dimension and generating a NumPy array of the image. The architectures took a long time to build the model. The system got extremely slow during the training phase. The most challenging part was to build a model that could identify different stages of Diabetic Retinopathy correctly.

PLAGIARISM RESULT

141-15-105

ORIGINALITY REPORT

13%

SIMILARITY INDEX

10%

INTERNET SOURCES

9%

PUBLICATIONS

7%

STUDENT PAPERS

PRIMARY SOURCES

1	towardsdatascience.com Internet Source	6%
2	"Pattern Recognition and Computer Vision", Springer Science and Business Media LLC, 2018 Publication	1%
3	dspace.daffodilvarsity.edu.bd:8080 Internet Source	1%
4	www.scribd.com Internet Source	1%
5	Submitted to Daffodil International University Student Paper	1%
6	endocrinology.endocrineconferences.com Internet Source	<1%
7	Gao Huang, Zhuang Liu, Geoff Pleiss, Laurens Van Der Maaten, Kilian Weinberger. "Convolutional Networks with Dense Connectivity", IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019	<1%

Publication

8	"Emerging Research in Electronics, Computer Science and Technology", Springer Science and Business Media LLC, 2019	<1%
Publication		
<hr/>		
9	Junya Kizawa, Shigeki Machida, Takaki Kobayashi, Yasutaka Gotoh, Daijiro Kurosaka. "Changes of Oscillatory Potentials and Photopic Negative Response in Patients with Early Diabetic Retinopathy", Japanese Journal of Ophthalmology, 2006	<1%
Publication		
<hr/>		
10	en.wikipedia.org	<1%
Internet Source		
<hr/>		
11	Mookiah, M.R.K., U. Rajendra Acharya, Roshan Joy Martis, Chua Kuang Chua, C.M. Lim, E.Y.K. Ng, and Augustinus Laude. "Evolutionary algorithm based classifier parameter tuning for automatic diabetic retinopathy grading: A hybrid feature extraction approach", Knowledge-Based Systems, 2013.	<1%
Publication		
<hr/>		
12	Submitted to University of Teesside	<1%
Student Paper		
<hr/>		
13	www.sciencedirect.com	<1%
Internet Source		

14	Submitted to TechKnowledge Student Paper	<1%
15	nptel.ac.in Internet Source	<1%
16	journals.plos.org Internet Source	<1%
17	export.arxiv.org Internet Source	<1%
18	Submitted to Sim University Student Paper	<1%
19	indianahimss.org Internet Source	<1%

Exclude quotes On

Exclude matches < 10 words

Exclude bibliography On