DIABETIC RETINOPATHY DETECTION

Submitted in partial fulfillment of the requirements

of the degree of

Bachelor of Engineering

in

Information Technology

by

Nishi Agrawal

Aakash Sinha

Kaushik Parmar

Supervisor:

Prof. Jigar Chauhan

Vidya Vikas Eductation Trust’s

Universal College Of Engineering

Accredited With B+ Grade by NAAC

Near Bhajansons and Punyadham, Kaman Bhiwandi Road,
Thane, Vasai-401212

2019-2020

CERTIFICATE

This is to certify that the project entitled **”Diabetic Retinopathy Detection”** is a bonafide work of **“Nishi Agrawal(1), Aakash Sinha(47), Kaushik Parmar(29)”** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **Undergraduate** in **”Information Technology”**

(Name and Sign)
Supervisor/Guide

(Name and sign)
Co-Supervisor/Guide

(Name and Sign)
Head of Department

(Name and sign)
Principal

Project Report Approval for B. E

The project report entitled Diabetic Retinopathy Detection by Nishi Agrawal, Aakash Sinha, Kaushik Parmar is approved for the degree of Bachelor Of Information Technology.

Examiners:

1.____________________________

2.____________________________

Date:

Place:

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Name of student and Roll no.)

(Name of student and Roll no.)

(Name of student and Roll no.)

Date:

Abstract

Diabetic Retinopathy (DR) is the leading cause of blindness in the working-age population of the developed world and is estimated to affect over 93 million people.Detecting DR is a time-consuming and manual process that requires a trained clinician to examine and evaluate digital color fundus photographs of the retina.In this report, we have proposed three different methods for classifying DR Images. The first method uses Convolutional Neural Network. The Second method uses a pre-trained ConvNet model for feature extraction. The damage in the retinal blood vessel eventually blocks the light that passes throughthe optical nerves which makes the patient with Diabetic Retinopathy blind. Therefore, in ourresearch we wanted to find out a way to over- come this problem and thus using the help ofconvolutional neural network (ConvNet), we were able to detect multiple stages of severity forDiabetic Retinopathy.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Diabetic retinopathy (DR), also known as diabetic eye disease, is when damage occurs to the retina due to diabetes. It can eventually lead to blindness. It is an ocular manifestation of diabetes. Despite these intimidating statistics, research indicates that at least 90 percent of these new cases could be reduced if there were proper and vigilant treatment and monitoring of the eyes. The longer a person has diabetes, the higher his or her chances of developing diabetic retinopathy. Diabetic retinopathy can be diagnosed into 5 stages: mild, moderate, severe, proliferative or no disease. The various signs and markers of diabetic retinopathy include microaneurysms, leaking blood vessels, retinal swellings, growth of abnormal new blood vessels and damaged nerve tissues. DR detection is challenging because by the time human readers submit their reviews, often a day or two later, the delayed results lead to lost follow up, miscommunication, and delayed treatment. Clinicians can identify DR by the presence of lesions associated with the vascular abnormalities caused by the disease. While this approach is effective, its resource demands are high. The expertise and equipment required are often lacking in areas where the rate of diabetes in local populations is high and DR detection is most needed. The need for a comprehensive and automated method of DR screening has long been recognized, and previous efforts have made good progress using image classification, pattern recognition, and machine learning. The current research in diagnosing diabetic retinopathy has been based on explicit extraction of features like microaneurysms and lesions through which the classification is performed. There has also been research in using machine learning techniques to classify the image as normal or diseased. This model aims at proposing a diabetic retinopathy diagnosis model that automatically learns features which are pivotal in diagnosing the stage of the disease without explicit or manual feature extraction Diabetic Retinopathy (DR) is one of the major causes of blindness in the western world. Increasing life expectancy, indulgent lifestyles and other contributing factors mean the number of people with diabetes is projected to continue rising . Regular screening of diabetic patients for DR has been shown to be a cost-effective and important aspect of their care. The accuracy and timing of this care is of significant importance to both the cost and effectiveness of treatment. If detected early enough, effective treatment of DR is available, making this a vital process. Classification of DR involves the weighting of numerous features and the location of such features. This is highly time consuming for clinicians. Computers are able to obtain much quicker classifications once trained, giving the ability to aid clinicians in real-time classification. The efficacy of automated grading for DR has been an active area.

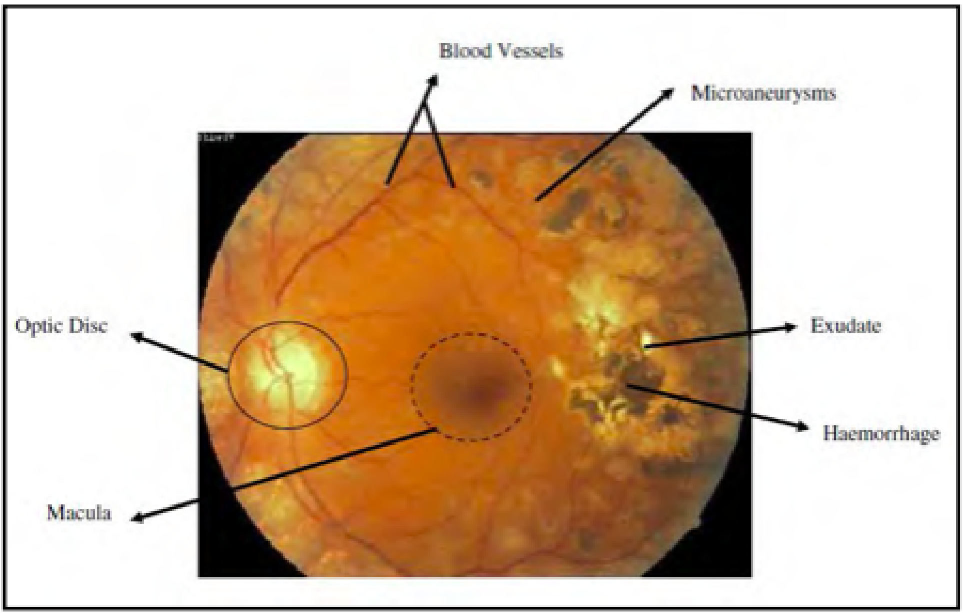


Figure 1.1: Features in DR image

1.1 Motivation

The main motivation behind the creation of this project is to create an ideal system to detect diabetic retinopathy at the earliest stage. Another vision is to create a system which has the maximum accuracy.

1.2 Problem Statement

Diabetic Retinopathy is the leading cause of blindness in the working-age population of the developed world. It is estimated to affect over 93 million people. The US center for disease control and prevention estimates that 29.1 million people in the US have diabetes and the World Health Organization estimates that 347 million people have the disease worldwide. Around 40 percent to 45 percent of Americans with diabetic have some stage of this disease. Progress to vision impairment can be slowed or averted if Diabetic Retinopathy is detected in time, however this can be difficult as the disease often shows few symptoms until it is too late to provide effective treatment. Currently, detecting Diabetic Retinopathy is a time consuming and manual process that requires a trained clinician to examine and evaluate digital colour fundus photographs of the retina.

1.3 Scope of monitoring

In today's world as there are everything is digital. In manual system work is very lengthy and time consuming and also required extra manpower. We develop this software for doctors, hospitals and patients. In this project we implement our proposed system, due to this the patient does not have to wait for the results. This project en- ables to gain maximum accuracy for the current model. Another scope of this project is to implement Diabetic Retinopathy detection processing model using new algorithms. The results will be attained in real time. Diabetic Retinopathy project will be built using artificial intelligence, machine learning as well as image pre-processing. This system will enable us to replace the doctor intervention in the detection of Diabetic Retinopathy.

1.4 Constraints of the Project

1. Use of high quality images which will result in higher accuracy of the model is not possible due to non availability of high quality images

2. As the accuracy is low despite trying various algorithms, a professional will be needed to review the scans

Chapter 2

Literature Review

Title	Description	Advantages	Disadvantages
106-112, Automated detection of diabetic retinopathy using SVM	They present a series of experiments on feature selection and CLASSIFICATION	simple algorithms like kmean's and svm	accuracy is very low .
18-24, Diabetic retinopathy: An epidemic at home and around the world	after preprocess,morphol ogical operations are performed to find the feature and gets as glcm	Sensitivity and specificity of 87 and 100	High and complex computing required
86 558-561,An Improved Approach for Detection of Diabetic Retinopathy Using Feature Importance and Machine Learning Algorithms	A quick automated detection system is implemented.	Machine learning Algorithm uses eight different classifications	it is costly in terms of time and might affect the patients

Table 2.1: Literature Survey Table

Chapter 3

Proposed System

3.1 Proposed System

Convolutional Neural Network-Based Image Classification The convolutional neural network (CNN) belongs to the feed-forward artificial neural network (ANN), which is very similar to ordinary neural networks. The CNN is a well-known deep learning architecture, in which individual neurons are tiled in such a way that they respond to overlapping regions in the visual fields . CNNs are an important class of learnable representation applications, and they were inspired by biological neural networks. Numerous variants have been proposed over last several years. However, the basic components are very similar. CNNs consist of alternating convolution and pooling operations . Typically, the convolutional layers are interspersed with pooling layers to reduce computation time, and build up further spatial and configuration invariance [?]; the last few layers (close to the outputs) will be fully connected 1-dimensional layers. In more detail, a feed-forward neural network can be viewed as a function f of mapping data x: f(x) = fL(... f2(f1(x1, w1), w2)..., wL). (1) Each function f l takes xl (x1 is the input data x) as input with a learnable parameter vector wl . L denotes the depth of the neural network. Although the type and sequence of functions are usually handcrafted, the parameters can be discriminatively learned from example data such that the resulting function f realizes a useful mapping. Formally, in a CNN, each xl will be a M N C array. As our problem can be simplified as a binary-classification problem, we can define the loss function of the CNN as: L(w) = 1/n (n i=1 l(zi , f(xi ; w))), (2) where n is the number of samples, zi is the true label of sample i. The training problem can be converted to training a neural network to minimize the loss function L. Figure ?? gives the general architecture of a CNN network, which consists of multiple layers of small neurons. The results of these collections are Molecules 2017, 22, 2054 4 of ?? then tiled so that they overlap to obtain a better representation of the original image (such as edges in the image). Convolutional layers consist of a rectangle grid of neurons, which takes a rectangle region of the previous layer as input. Moreover, there may be several grids in each convolutional layer, using potentially different filters. Typically, there is a pooling layer after each convolutional layer, which are subsampled from the previous convolutional layer. This pooling can be carried out in several ways, such as the average, maximum, etc. Finally, after several convolutional layers and max pooling layers, a fully connected layer (or several layers) will be built using outputs from previous layers (maybe the fully connected, pooling or convolutional layer), which is used as a compact feature to describe the whole input image. The network is optimized by backpropagation and stochastic gradient descent. Note that the forward and backward propagations may differ depending on the type of the layer

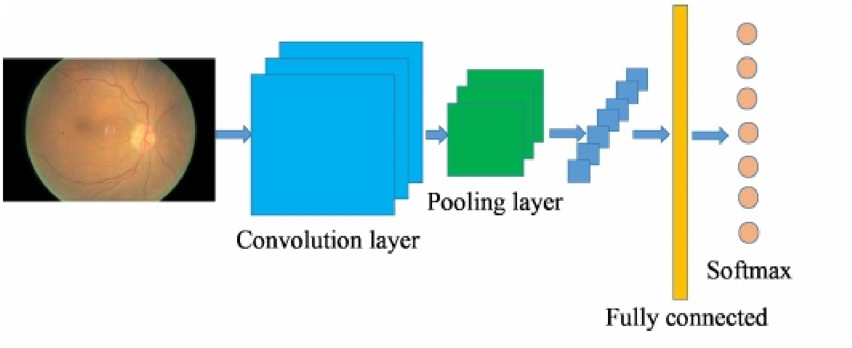
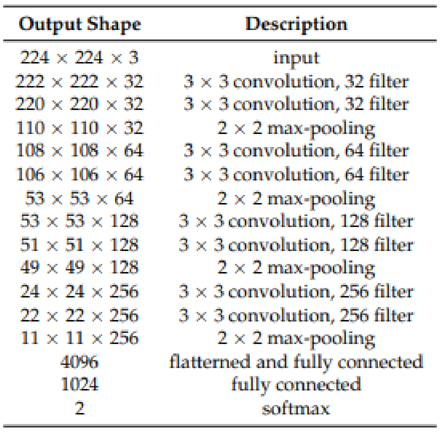


Figure 3.1: Segrergated Layers



Output Shape	Description
224 × 224 × 3	input
222 × 222 × 32	3 × 3 convolution, 32 filter
220 × 220 × 32	3 × 3 convolution, 32 filter
110 × 110 × 32	2 × 2 max-pooling
108 × 108 × 64	3 × 3 convolution, 64 filter
106 × 106 × 64	3 × 3 convolution, 64 filter
53 × 53 × 64	2 × 2 max-pooling
53 × 53 × 128	3 × 3 convolution, 128 filter
51 × 51 × 128	3 × 3 convolution, 128 filter
49 × 49 × 128	2 × 2 max-pooling
24 × 24 × 256	3 × 3 convolution, 256 filter
22 × 22 × 256	3 × 3 convolution, 256 filter
11 × 11 × 256	2 × 2 max-pooling
4096	flatterned and fully connected
1024	fully connected
2	softmax

Figure 3.2: CNN layers

3.2 System Overview

3.2.1 Dataset

The dataset consists of 35,126 labelled high-resolution colour fundus retinal images belonging to five classes corresponding to the five stages of the disease as portrayed in Table I. The test set consists of 250 images which have been utilized for this paper. The images have been open-sourced by EyePACs, a free platform for retinopathy screening. A trained clinician has rated the presence of diabetic retinopathy in each image on a scale of 0 to 4 [5]. The images in the dataset come from different models and types of cameras, which can affect the visual appearance of left and right retinas. Some images are shown as one would see the retina anatomically (macula on the left, optic nerve on the right for the right eye). Others are shown as one would see through a microscope condensing lens (i.e. inverted, as one sees in a typical live eye exam). There is also noise in both the images and labels. Images may contain artifacts, be out of focus, underexposed, or overexposed and are of different resolutions. Table 1 shows the distribution of image classes in the dataset.

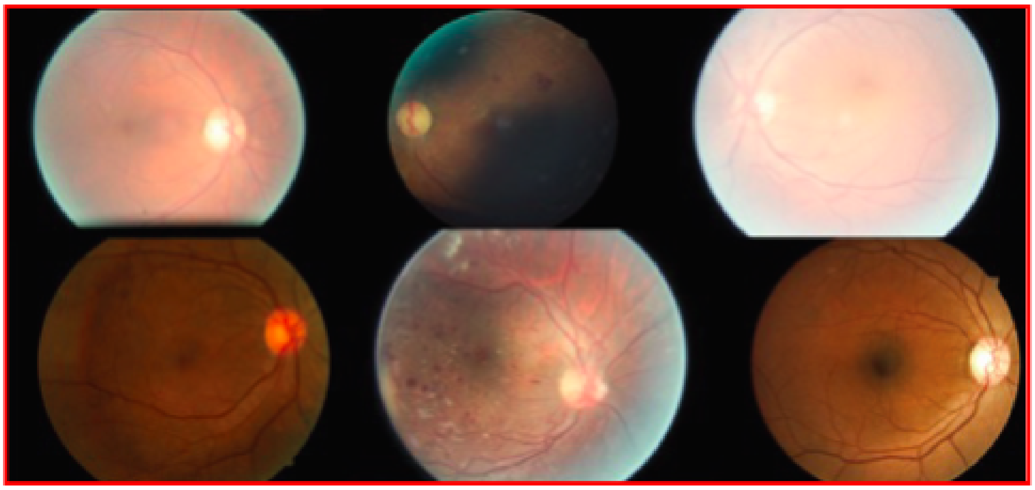
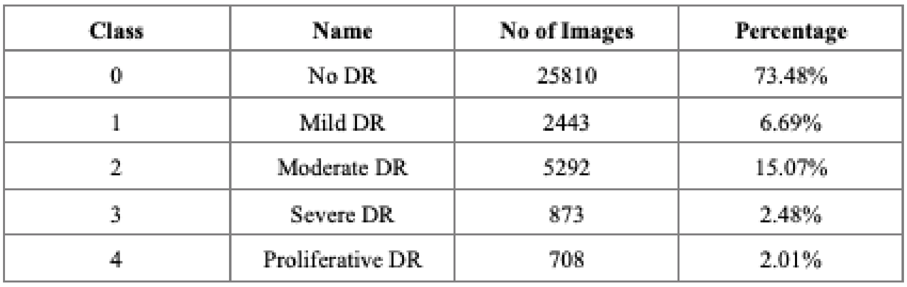


Figure 3.3: Sample DR Image



Class	Name	No of Images	Percentage
0	No DR	25810	73.48%
1	Mild DR	2443	6.69%
2	Moderate DR	5292	15.07%
3	Severe DR	873	2.48%
4	Proliferative DR	708	2.01%

Figure 3.4: Class Distribution in Original Dataset

3.2.2 Dataset Pre-Processing

Data Pre-Processing Due to non-standard image resolutions, the training images could not be utilized directly for training. The images were scaled down to a fixed resolution size of 512x512 pixels to form a standardized dataset. Training images of resolution 512x512 pixels on all three colour channels demanded high memory requirements. Due to this limitation, the images were converted to a single channel. After several experiments, it was found that green channel images retained information better than the other channel images. In order to enhance the contrast of the image evenly across pixels, histogram equalization technique was applied on the images. In order to prevent the convolutional neural network from learning the inherent background noise in the image, each image was normalized using Min-Max normalization. The network contains an input layer which takes images with resolution 512x512 pixels as input. The architecture consists of 5 sets of combination of convolution, pooling and dropout layers, each stacked one over the other. This is followed by 2 sets of fully connected hidden and feature pooling layers. This is followed by the final output layer. To preserve the spatial size of the input and output volumes, zero-padding was utilized in the convolutional layers.

3.2.3 Machine Learning

Machine learning, a branch of artificial intelligence, concerns the construction and study of systems that can learn from data . Machine learning algorithms use computational methods to “learn” information directly from data without relying on a predetermined equation as a model. The algorithms adaptively improve their performance as the number of samples available for learning increases. Tom M. Mitchell provided a widely quoted and more formal definition: A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E. The core of machine learning deals with representation and generalization. Representing the data instances and functions evaluated on these instances are part of all machine learning systems. Generalization is the ability of a machine learning system to perform accurately on new, unseen data instances after having experienced a learning data instance. The training examples come from some generally unknown probability distribution and the learner has to build a general model about this space that enables it to produce sufficiently accurate predictions in new cases. The performance of generalization is usually evaluated with respect to the ability to reproduce known knowledge from newer examples.

3.3 Design Considerations

Data Flow Diagram

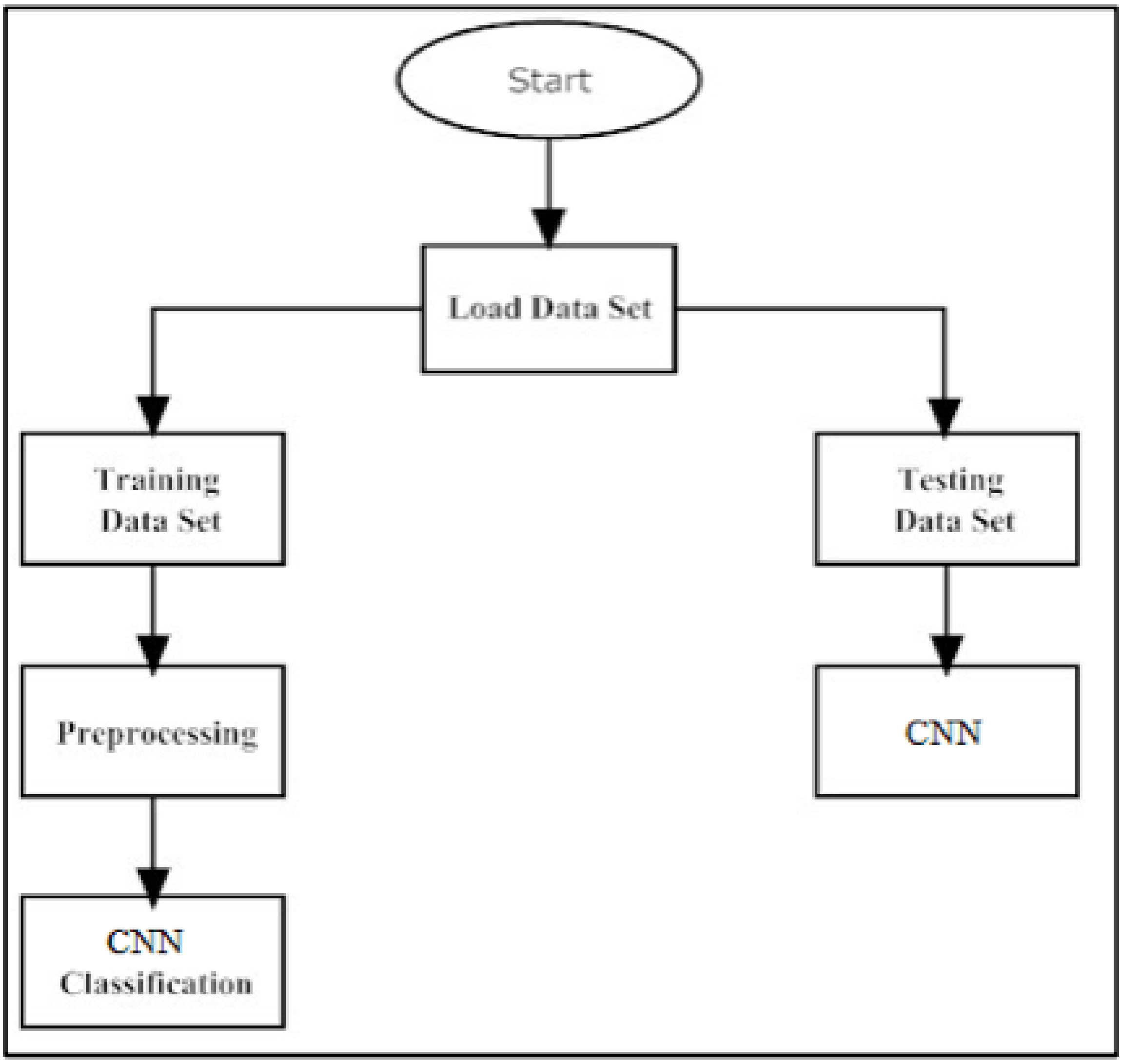


Figure 3.5: Data Flow Diagram

Sequence Diagram

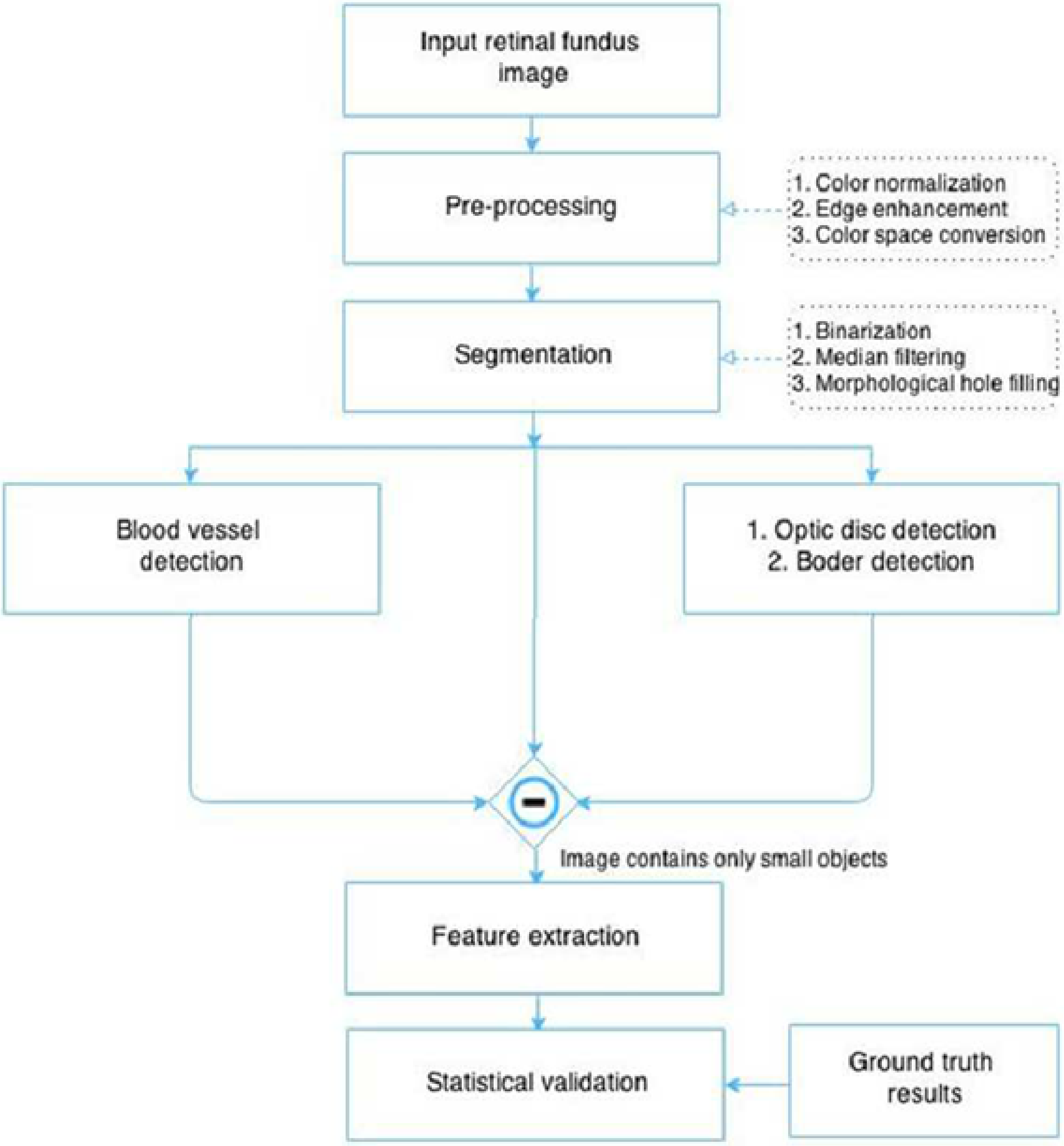


Figure 3.6: Sequence Diagram

Activity Diagram

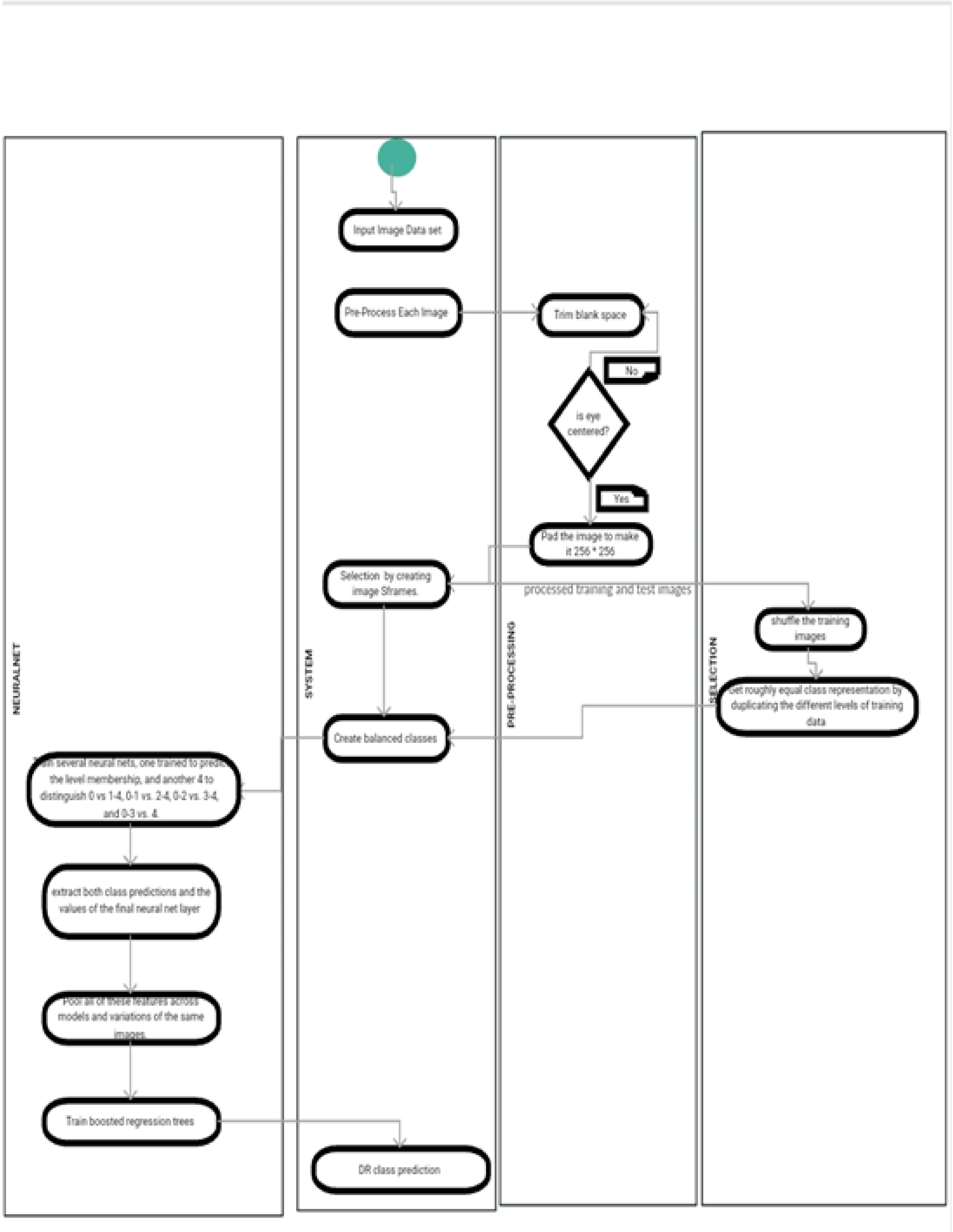

NEURALNET
SYSTEM
PRE-PROCESSING
SELECTION
Input Image Data set
Pre-Process Each Image
Trim blank space
No
is eye centered?
Yes
Pad the image to make it 256 * 256
Selection by creating image Sframes.
processed training and test images
shuffle the training images
Get roughly equal class representation by duplicating the different levels of training data
Create balanced classes
ain several neural nets, one trained to predic the level membership, and another 4 to distinguish 0 vs 1-4, 0-1 vs. 2-4, 0-2 vs. 3-4, and 0-3 vs. 4.
extract both class predictions and the values of the final neural net layer
Pool all of these features across models and variations of the same images.
Train boosted regression trees
DR class prediction

Figure 3.7: Activity Diagram

Class Diagram

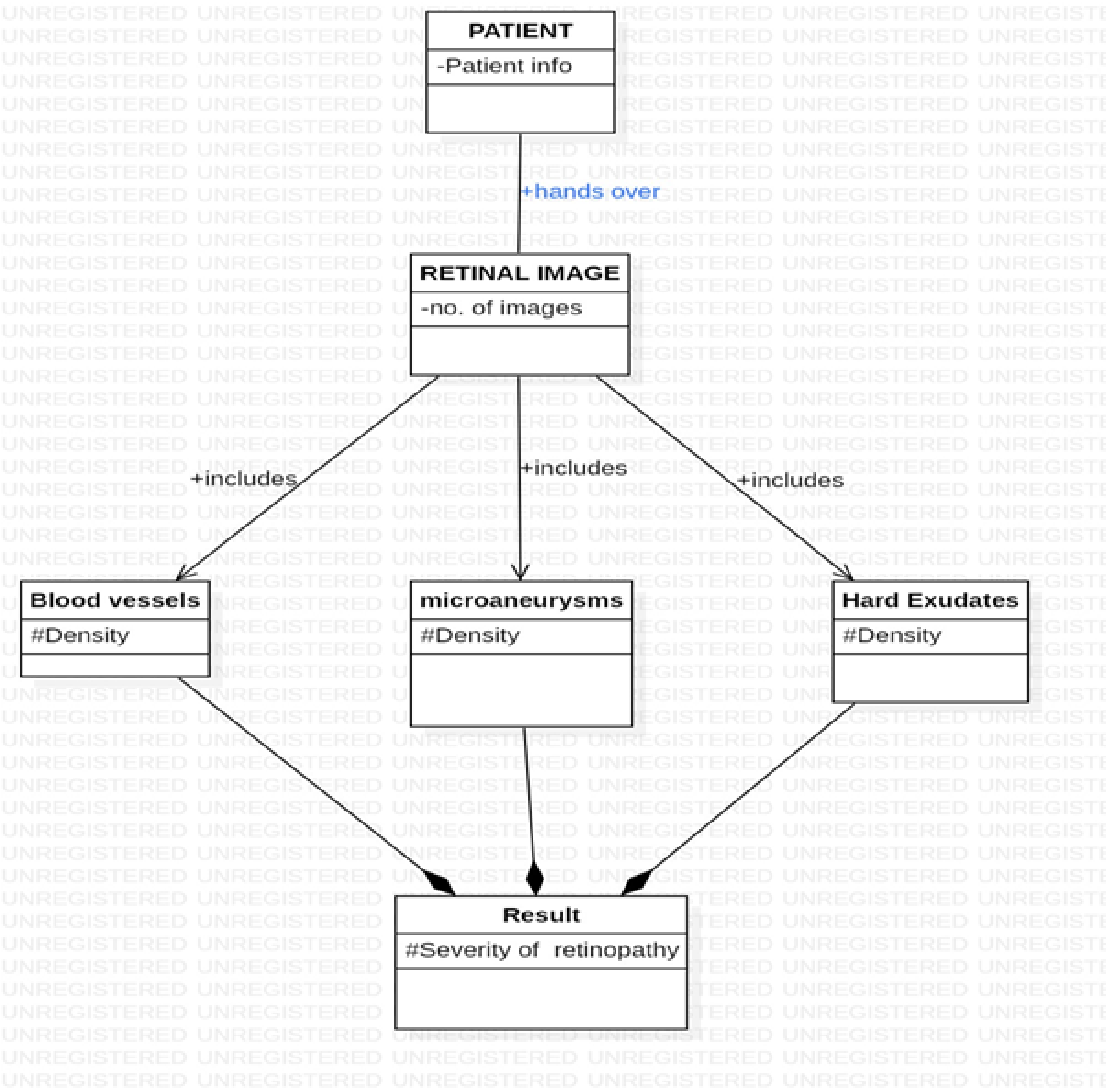


Figure 3.8: Class Diagram

3.4 Flowchart

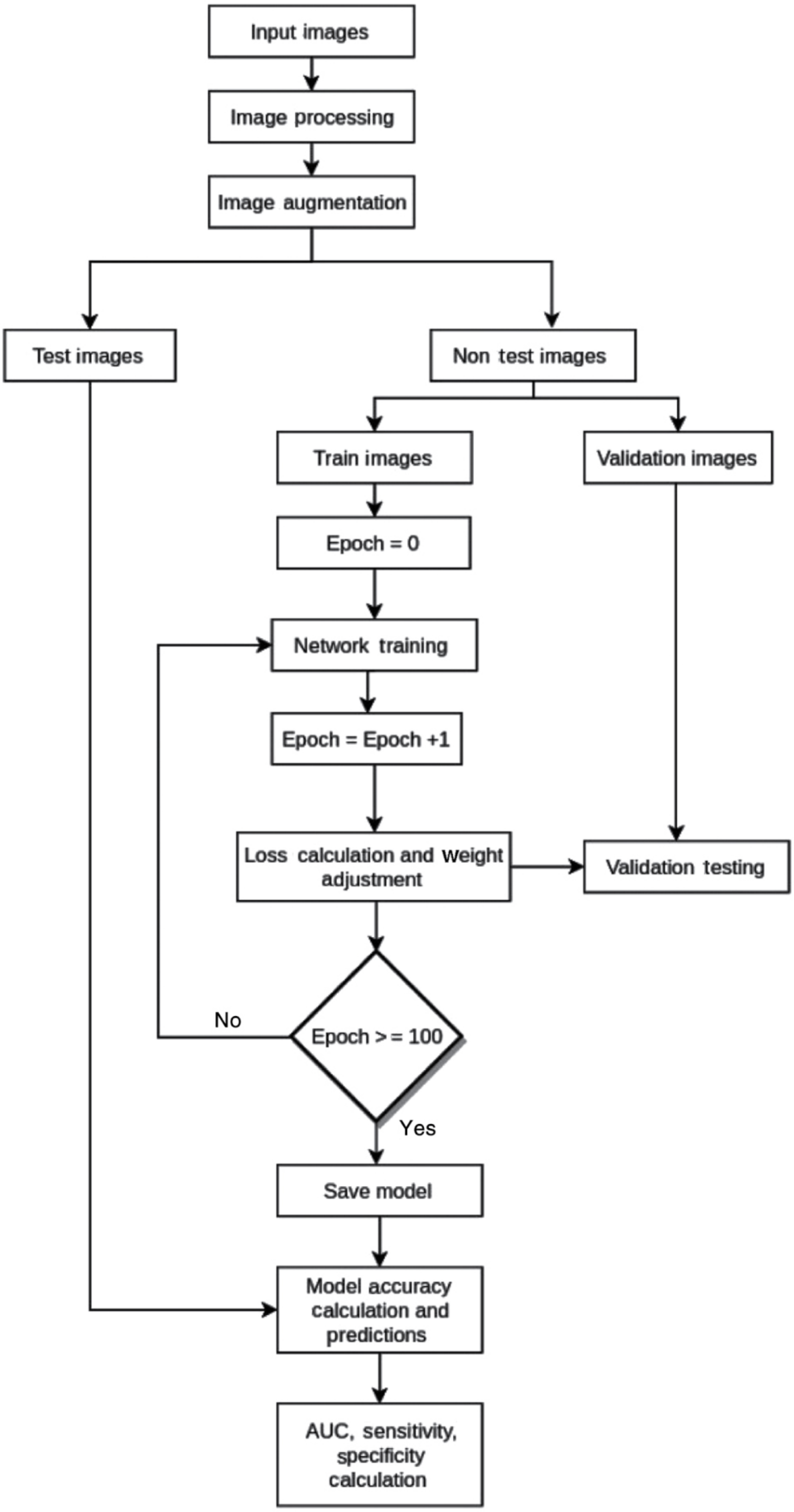


Figure 3.9: System Flowchart

3.5 Hardware Implementation

3.5.1 4 GB RAM

The maximum random access memory (RAM) installed in any computer system is limited by hardware, software and economic factors. The hardware may have a limited number of address bus bits, limited by the processor package or design of the system. Some of the address space may be shared between RAM, peripherals, and read-only memory. In the case of a microcontroller with no external RAM, the size of the RAM array is limited by the size of the integrated circuit die. In a packaged system, only enough RAM may be provided for the system's required functions, with no provision for addition of memory after manufacture.

3.5.2 10 GB hard drive

A hard disk drive (HDD), hard disk, hard drive, or fixed disk[b] is an electro-mechanical data storage device that uses magnetic storage to store and retrieve digital data using one or more rigid rapidly rotating platters coated with magnetic material. The platters are paired with magnetic heads, usually arranged on a moving actuator arm, which read and write data to the platter surfaces.[2] Data is accessed in a random-access manner, meaning that individual blocks of data can be stored and retrieved in any order. HDDs are a type of non-volatile storage, retaining stored data even when powered off

3.5.3 Intel 1.66GHZ

A processor (CPU) is the logic circuitry that responds to and processes the basic instructions that drive a computer. The CPU is seen as the main and most crucial integrated circuitry (IC) chip in a computer, as it is responsible for interpreting most of computers commands.

3.6 Software Implementation

3.6.1 Python

Python is an interpreted, high-level, general-purpose programming language Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.[. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented, and functional programming. Python is often described as a ”batteries included” language due to its comprehensive standard library

3.6.2 Keras Module

Keras is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, R, Theano, or PlaidML. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible. It was developed as part of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System) and its primary author and maintainer is François Chollet, a Google engineer. Keras contains numerous implementations of commonly used neural-network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier to simplify the coding necessary for writing deep neural network code

3.6.3 Google Colab

Colaboratory, or “Colab” for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education. More technically, Colab is a hosted Jupyter notebook service that requires no setup to use, while providing free access to computing resources including GPUs.Colab is free to useJupyter is the open source project on which Colab is based. Colab allows you to use and share Jupyter notebooks with others without having to download, install, or run anything.Colab notebooks are stored in Google Drive, or can be loaded from GitHub. Colab notebooks can be shared just as you would with Google Docs or Sheets. Simply click the Share button at the top right of any Colab notebook.Code is executed in a virtual machine private to your account. Virtual machines are deleted when idle for a while, and have a maximum lifetime enforced by the Colab service.

Chapter 4

Implementation

Data Collection

In our project we have used a dataset that is obtained from the UCI Machine Learning Repository. This dataset contains features extracted from Messidor image set to predict whether an image contains signs of diabetic retinopathy or not. All features represent either a detectedv lesion, a descriptive feature of an anatomical part or an image-level descriptor. The Messidor database has been established to facilitate studies on computer-assisted diagnoses of diabetic retinopathy. We have seen different kind of datasets in kaggle, github and other websites which was used for different kind of projects based on diabetic retinopathy. As we wanted to work with detection of diabetic retinopathy, this dataset will be appropriate for our work as it has different types of features.

Data Description

Our dataset contains different types of features that is extracted from the Messidor image set. This dataset is used to predict whether an image contains signs of diabetic retinopathy or not. The value here represents different point of retina of diabetic patients. First 19 columns in the dataset are independent variables or input column and last column is dependent variables or output column. Outputs are represented by binary numbers. “1” means the patient has diabetic retinopathy and “0” means absence of the disease.

Data Visualization

Another important feature in the data distribution is the skewness of each class. Data visualization helps to see how the data looks like and also what kind of data correlation we have. The dataset distribution of each feature is shown below in figure 3.5. This is a histogram. A histogram is an accurate graphical representation of the distribution of numerical data. It is an estimate of the probability distribution of a continuous variable. Histograms are a great way to get to know your data. They allow you to easily see where a large and a little amount of the data can be found. In short, the histogram consists of an x-axis and a y-axis, where the y-axis shows how frequently the values on the x-axis occur in the data.

Split Dataset

Separating data into training and testing sets is an important part of evaluating data mining models. Typically, when separating a data set into two parts, most of the data is used for training, and a smaller portion of the data is used for testing. We have also split our dataset into two sets. One is for training and another for testing. The training set contains a known output and the model learns on this data in order to be generalized to other data later on. After the model has been processed by using the training set, we have tested the model by making predictions against the test set. Because the data in the testing set already contains known values for the attribute that we want to predict, it is easy to determine whether the model's guesses are correct or not. In addition, we have used 80

Applying Algorithm

We went through a process of trial and error to settle on a short list of algorithms that provides better result as we are working on classification of diabetic retinopathy, we used some machine learning classification algorithms. We get an idea from the data visualizations plots which algorithms will be suitable for the classification problem. The Machine Learning system uses the training data to train models to see patterns, and uses the test data to evaluate the predictive quality of the trained model. Machine learning system evaluates predictive performance by comparing predictions on the evaluation data set with true values (known as ground truth) using a variety of metrics.

Training Phase

Classification assumes labeled data: we know how many classes there are, and for each class we have examples (labeled data).

Testing Phase

Testing phase involves the prediction of unknown data sample.

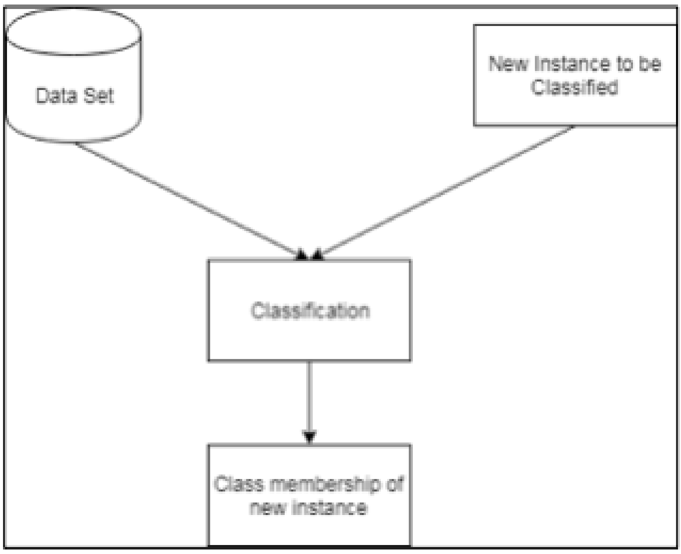

Data Set
New Instance to be Classified
Classification
Class membership of new instance

Figure 4.1: Training Phase

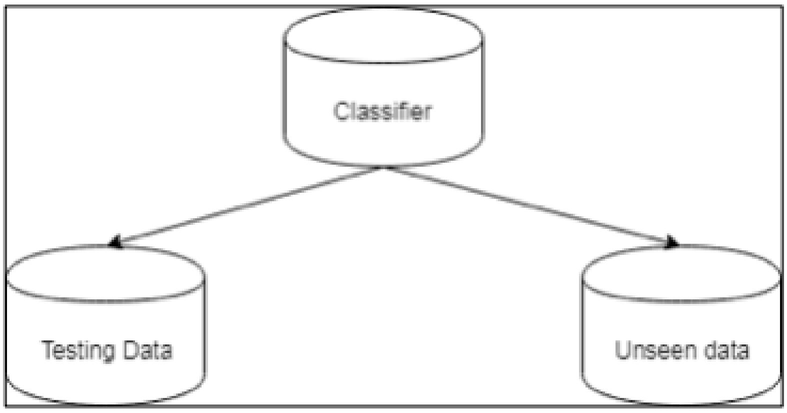


Figure 4.2: Testing Phase

Chapter 5

Testing

5.1 Unit Testing

Unit testing refers to numerically asserting the expected behavior of each function. Each layer type only contains two testing-worthy functions: Forward and backward propagation. Let's make that clear with a concrete example: We start by defining the expected behavior. The max pooling layer slides a kernel over its input. At each stage it selects the neuron with the maximum value within the current receptive field and assigns it output Library used in our project have to support every valid configuration, therefore these should also be covered by our testing suite. In a realistic scenario we would expect the input to span multiple feature maps, the stride to be higher than 1 and maybe not even symmetrical between the vertical and horizontal dimensions. The padding could be adapted to control the output sizeThe same process is then followed for both the forward and back propagation of all supported layer types.

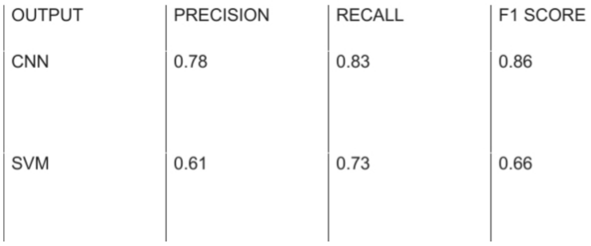
5.2 Integration Testing

We assign arbitrary input and weight values for each layer of a neural network. That network should include at least one layer of each type. The depths must be greater than 1 and the strides asymmetrical to make sure we are checking all cases. Then propagate the input through the network and compute the output of each layer on paper. Warning bells start ringing for most sane people at this step. The last step would entail performing back-propagation on paper, by computing the activation and weight gradients on paper for each layer.

Chapter 6

Result Analysis

We compare our results with the deep learning method proposed by Automatic detection and classification of diabetic retinopathy stages using CNN We are getting better results transfer learning from Automatic detection and classification of diabetic retinopathy stages using CNN than with other algorithm like svm. Our model gives correct classification for 58 percent of the time while their[1] proposed model gives correct classification 36 percent of the time .



OUTPUT	PRECISION	RECALL	F1 SCORE
CNN	0.78	0.83	0.86
SVM	0.61	0.73	0.66

Figure 6.1: CNN SVM output

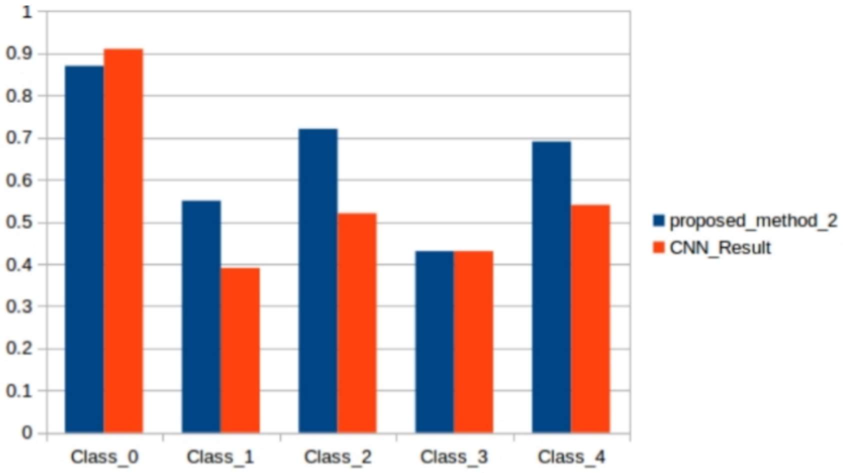
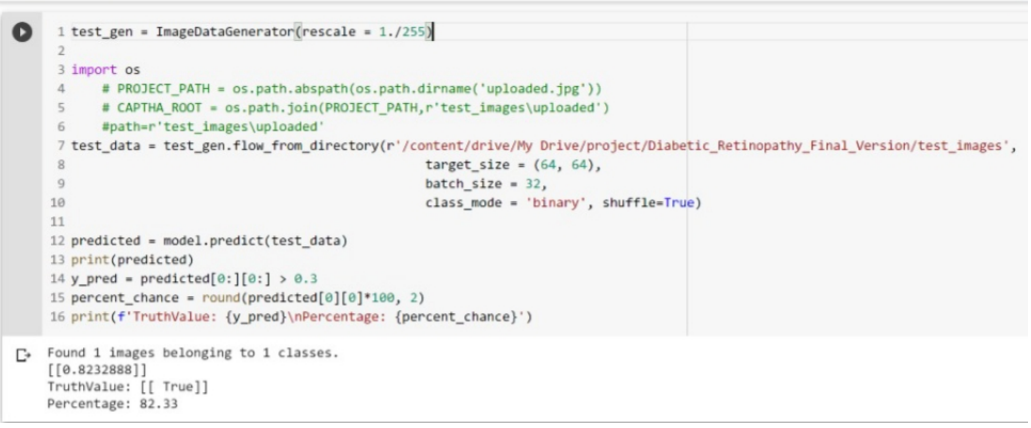


Figure 6.2: f1score comparison

Chapter 7

Results and Discussions

7.1 Screenshots of the Working Project



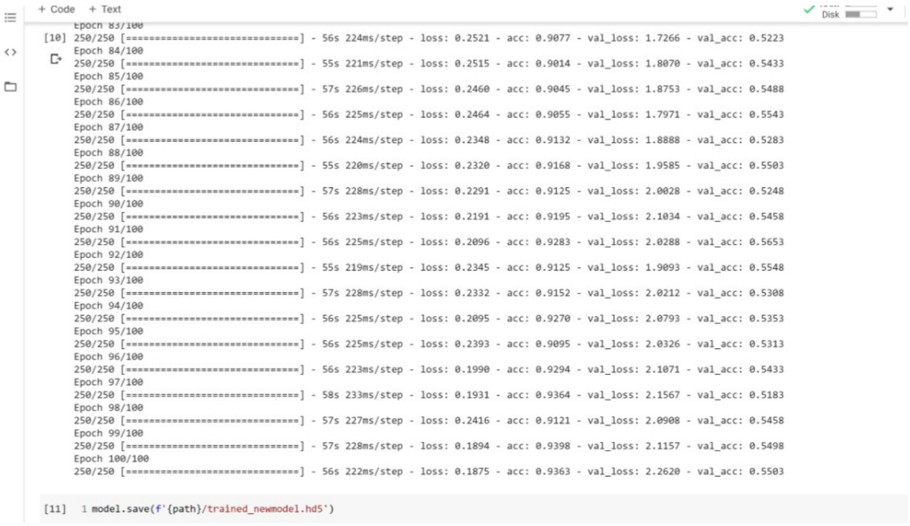
```
test_gen = ImageDataGenerator(rescale = 1./255)

import os
    # PROJECT_PATH = os.path.abspath(os.path.dirname('uploaded.jpg'))
    # CAPTHA_ROOT = os.path.join(PROJECT_PATH,r'test_images\uploaded')
    #path=r'test_images\uploaded'
test_data = test_gen.flow_from_directory(r'/content/drive/My Drive/project/Diabetic_Retinopathy_Final_Version/test_images',
                                          target_size = (64, 64),
                                          batch_size = 32,
                                          class_mode = 'binary', shuffle=True)

predicted = model.predict(test_data)
print(predicted)
y_pred = predicted[0:][0:] > 0.3
percent_chance = round(predicted[0][0]*100, 2)
print(f'TruthValue: {y_pred}\nPercentage: {percent_chance}')
```

```
Found 1 images belonging to 1 classes.
[[0.8232888]]
TruthValue: [[ True]]
Percentage: 82.33
```

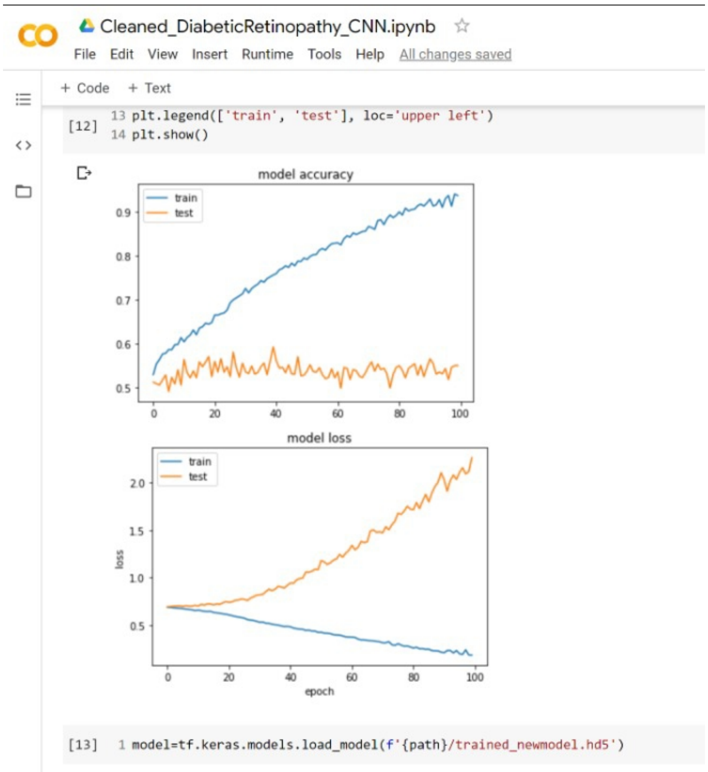
Figure 7.1: Image Input



```
Epoch 83/100
[10] 250/250 [==============================] - 56s 224ms/step - loss: 0.2521 - acc: 0.9077 - val_loss: 1.7266 - val_acc: 0.5223
Epoch 84/100
250/250 [==============================] - 55s 221ms/step - loss: 0.2515 - acc: 0.9014 - val_loss: 1.8070 - val_acc: 0.5433
Epoch 85/100
250/250 [==============================] - 57s 226ms/step - loss: 0.2460 - acc: 0.9045 - val_loss: 1.8753 - val_acc: 0.5488
Epoch 86/100
250/250 [==============================] - 56s 225ms/step - loss: 0.2464 - acc: 0.9055 - val_loss: 1.7971 - val_acc: 0.5543
Epoch 87/100
250/250 [==============================] - 56s 224ms/step - loss: 0.2348 - acc: 0.9132 - val_loss: 1.8888 - val_acc: 0.5283
Epoch 88/100
250/250 [==============================] - 55s 220ms/step - loss: 0.2320 - acc: 0.9168 - val_loss: 1.9585 - val_acc: 0.5503
Epoch 89/100
250/250 [==============================] - 57s 228ms/step - loss: 0.2291 - acc: 0.9125 - val_loss: 2.0028 - val_acc: 0.5248
Epoch 90/100
250/250 [==============================] - 56s 223ms/step - loss: 0.2191 - acc: 0.9195 - val_loss: 2.1034 - val_acc: 0.5458
Epoch 91/100
250/250 [==============================] - 56s 225ms/step - loss: 0.2096 - acc: 0.9283 - val_loss: 2.0288 - val_acc: 0.5653
Epoch 92/100
250/250 [==============================] - 55s 219ms/step - loss: 0.2345 - acc: 0.9125 - val_loss: 1.9093 - val_acc: 0.5548
Epoch 93/100
250/250 [==============================] - 57s 228ms/step - loss: 0.2332 - acc: 0.9152 - val_loss: 2.0212 - val_acc: 0.5308
Epoch 94/100
250/250 [==============================] - 56s 225ms/step - loss: 0.2095 - acc: 0.9270 - val_loss: 2.0793 - val_acc: 0.5353
Epoch 95/100
250/250 [==============================] - 56s 225ms/step - loss: 0.2393 - acc: 0.9095 - val_loss: 2.0326 - val_acc: 0.5313
Epoch 96/100
250/250 [==============================] - 56s 223ms/step - loss: 0.1990 - acc: 0.9294 - val_loss: 2.1071 - val_acc: 0.5433
Epoch 97/100
250/250 [==============================] - 58s 233ms/step - loss: 0.1931 - acc: 0.9364 - val_loss: 2.1567 - val_acc: 0.5183
Epoch 98/100
250/250 [==============================] - 57s 227ms/step - loss: 0.2416 - acc: 0.9121 - val_loss: 2.0908 - val_acc: 0.5458
Epoch 99/100
250/250 [==============================] - 57s 228ms/step - loss: 0.1894 - acc: 0.9398 - val_loss: 2.1157 - val_acc: 0.5498
Epoch 100/100
250/250 [==============================] - 56s 222ms/step - loss: 0.1875 - acc: 0.9363 - val_loss: 2.2620 - val_acc: 0.5503

[11] model.save(f'{path}/trained_newmodel.hd5')
```

Figure 7.2: Generating Epochs



```
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```

```
model=tf.keras.models.load_model(f'{path}/trained_newmodel.hd5')
```

Figure 7.3: Training Accuracy

7.2 Advantages

- Early detection
- Low capital set up
- Retinal image can be used in patient education
- Hard copy can be incorporated into patient record
- Amenable to audit
- Accessible, convenient service
- Offers holistic package of eye care to the patient
- Utilizes the well trained, motivated workforce that optometrists represent

7.3 Limitations

- Difficulties in reaching all patients who need to be screened
- Need to provide regular training for graders
- Requires an elaborate quality control mechanism for the system to be audited
- High end hardware technology is required

7.4 Applications

7.4.1 Emerging Image Modalities for DR Screening

The narrow field of view in ordinary fundus photographs (e.g., single-field 45 or even a 7-field standard ETDRS) may exclude peripheral DR lesions. This is significant when predominantly peripheral lesions (PPLs) are associated with retinal ischemia, DR progression, and development of proliferative retinopathy [58, 59]. Eyes with PPLs were shown to have a 3.2-fold increased risk of two-step or more DR progression and a 4.7-fold increased risk for progression to proliferative disease [58]. Ultrawide-field (UWF) imaging, which can image 80

7.4.2 OCT Angiography as a Screening Tool

OCT angiography (OCT-A) is a novel noninvasive imaging modality that can detect blood flow within the retina (Fig. 2). In diabetic retinopathy, microaneurysms and retinal neovascularization can be detected using OCT-A noninvasively [65]. Although the clinical utility of OCT-A remains to be established [66], several recent works have highlighted its potential for incorporation into DR screening approaches. Accurate noninvasive grading of DR severity by OCT-A analysis has been demonstrated via analyses of foveal avascular zone acircularity [67] and retinal plexus densitometry [68–70]. Automation of OCT-A image analysis has also been used for quantitative assessment of DR vascular changes that correlate with visual acuity and may be useful for monitoring DR progression [71].

7.4.3 Diabetic Retinopathy Screening Programs in Modern Ophthalmic Practice

over the past few decades, the global prevalence of diabetes in adults has nearly doubled, increasing from 4.7 in 1980 to 8.5 in 2014, with rapid increases in middle- and low-income countries [1]. The World Health Organization projects that diabetes will be the seventh leading cause of death worldwide by 2030 [1]. Among diabetic complications, the prevalence of diabetic retinopathy (DR), diabetic macular edema (DME), and proliferative diabetic retinopathy (PDR) among individuals with diabetes is approximately 35, 7, and 3–7 percents respectively

Chapter 8

Conclusion and Future Scope

8.1 Conclusions

Automated detection and screening gives a opportunity to prevent a big proportion of vision loss in our population. CNNs promise to power the large amounts of images that have been used for physician interpreted screening and learn from raw pixels. Using transfer learning we are getting better results, except class0, f1-score for all the classes is greater than or equal to their[1] f1-score.

8.2 Future Scope

For CapsNet(a nueral network function)[10] we use image size of (192 * 192). We increase the image size i.e more than (192 * 192) than we get resource exhausted error. So due to limited resources we did not check the CapsNet for image size more than (192 * 192). So by increasing the image size we can get better results.

Bibliography

[1] R. Ghosh, K. Ghosh, and S. Maitra. Automatic detection and classification of diabetic retinopathy stages using cnn. In 2017 4th International Conference on Signal Processing and Integrated Networks (SPIN), pages 550–554, Feb 2017. doi: 10.1109/SPIN.2017.8050011

[2] Rajendra Acharya U, Chua Kuang Chua, E. Y. Ng, Wenwei Yu, and Caroline Chee. Application of higher order spectra for the identification of diabetes retinopathy stages. J. Med. Syst., 32(6):481–488, December 2008. ISSN 0148-5598. doi: 10.1007/ s10916-008-9154-8. URL http://dx.doi.org/10.1007/s10916-008-9154-8.

[3] Karel Zuiderveld. Graphics gems iv. chapter Contrast Limited Adaptive Histogram Equalization, pages 474–485. Academic Press Professional, Inc., San Diego, CA, USA, 1994. ISBN 0-12-336155-9. URL http://dl.acm.org/citation.cfm?id= 180895.180940.

[4] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, Nov 1998. ISSN 0018-9219. doi: 10.1109/5.726791.

[5] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37, ICML'15, pages 448–456. JMLR.org, 2015. URL http://dl.acm.org/citation. cfm?id=3045118.3045167.

Acknowledgement

We have great pleasure in presenting our project on “Diabetic Retinopathy Detection”. We wish to express our sincere thanks and deep sense of gratitude to respected guide Assistant Prof. Jigar Chauhan, Department of Information Technology, Universal College of Engineering, for their depth and enlightening support with all of their kindness. Their constant guidance and willingness to share their vast knowledge made us understand this topic and its manifestations in great depths.

We would also like to thank Prof. Yogita Mane, Head of the Department, Information Technology, Universal College of Engineering, for their support and co-ordination.

We are also grateful to the campus Director, Dr. Jitendra Patil, for providing us with all the facilities that we required for the successful completion of this project.

Finally our deepest gratitude goes to our family and friends who have given us good support, encouragement and inspiration to complete my work.

Ms. Nishi Agrawal ,

Mr. Aakash Sinha,

Mr. Kaushik Parmar .