GLAUCOMA DETECTION AND CLASSIFICATION

Submitted in partial fulfillment of the requirements
of the degree of

B. E. Computer Engineering

By

Royce Dcunha	**37**
Aaron Rodrigues	**38**
Manisha Sahu	**43**
Cassandra Rodrigues	**54**

Supervisor:

Dr. Kavita Sonawane
Professor & HOD

Department of Computer Engineering
St. Francis Institute of Technology
(Engineering College)

University of Mumbai
2021-2022

CERTIFICATE

This is to certify that the project entitled **“Glaucoma Detection and Classification”** is a bonafide work of **“Royce Dcunha” (37), “Aaron Rodrigues” (38), “Manisha Sahu” (43), “Cassandra Rodrigues” (54)** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of B.E. in Computer Engineering

Dr. Kavita Sonawane

Guide

Dr. Kavita Sonawane	**Dr. Sincy George**
Head of Department	**Principal**

Project Report Approval for B.E.

This project report entitled (***Glaucoma Detection and Classification***) by (***Royce Dcunha, Aaron Rodrigues, Manisha Sahu, Cassandra Rodrigues***) is approved for the degree of ***B.E. in Computer Engineering.***

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.

Royce Dcunha (37)

Aaron Rodrigues (38)

Manisha Sahu (43)

Cassandra Rodrigues (54)

Date:

Abstract

Deep learning is an important technique for investigating medical images. Glaucoma is a chronic eye disease that results from visual nerve damage caused by intraocular pressure in the eye. It is one of the leading causes of blindness around the globe and if not detected early enough, it can lead to complete blindness. In the early stages of glaucoma, there are no symptoms of vision loss, but as it progresses, it may result in irreversible blindness. It is often associated with an accumulation of pressure within your eye. Glaucoma is common among families. It usually happens later in life. Diagnosis of glaucoma in the clinical environment includes intraocular pressure measurement, visual field testing, or examination of the optical disk of fundus images.

Usually, people have no symptoms, and if symptoms occur, it is around the end of the illness. The primary sign is generally a loss of lateral vision or peripheral vision. Although glaucoma cannot be prevented, it can be reduced in severity if discovered early. In addition, the number of ophthalmologists required for evaluation by direct examination becomes a limiting factor due to aging, population growth, physical inactivity, and obesity which contributes to increasing the risk of vision loss. However, in large-scale screening scenarios, these manual assessments are not precise, mostly in developing countries due to the insufficiency of trained experts and scarce modern imaging equipment.

In this paper, several models are being used to study glaucoma detection. The models chosen are: VGG19, VGG19+LSTM, InceptionV3, and InceptionV3+LSTM. Every model is being worked with K-fold cross-validation and data augmentation to overcome the limitation of a small dataset. The features extracted are used to classify the input image and are then projected to be either glaucomatous or normal. Finally, the values obtained for various performance evaluation parameters are compared. The ACRIMA dataset consists of 705 fundus images (396 glaucomatous and 309 normal images), out of which 632 images are for training, 73 are for testing, with a 90-10 split. The evaluation results of the VGG19 model are, accuracy: 91.78%, precision for normal class: 85%, precision for glaucomatous class: 100%, recall for normal class: 100%, recall for glaucomatous class: 100%, and F1 score for normal class: 92%, F1 score for glaucomatous class: 92%.

The evaluation results of the VGG19+LSTM model are, accuracy: 94.52%, precision for normal class: 90%, precision for glaucomatous class: 100%, recall for normal class: 100%, recall for glaucomatous class: 89%, and F1 score for normal class: 95%, F1 score for glaucomatous class: 94%. The evaluation results of the Inception v3 model are, accuracy: 94.03%, precision for normal class: 91%, precision for glaucomatous class: 97%, recall for normal class: 97%, recall for glaucomatous class: 92%, and F1 score for normal class: 94%, F1 score for glaucomatous class: 94%. The evaluation results of the Inception v3+LSTM model are, accuracy: 90.41%, precision for normal class: 88%, precision for glaucomatous class: 93%, recall for normal class: 95%, recall for glaucomatous class: 85%, and F1 score for normal class: 92%, F1 score for glaucomatous class: 89%.

Contents

List of Figures

List of Tables

List of Abbreviations

Sr. No.	Abbreviation	Expanded form
i	CNN	Convolution Neural Network
ii	VGG	Visual Geometry Group
iii	LSTM	Long Short-Term Memory
iv	DFD	Data Flow Diagram
v	JPG	Joint Photographic Experts Group
vi	CSV	Comma Separated Values
vii	GPU	Graphics Processing Unit
viii	ML	Machine Learning
ix	NN	Neural Network
x	OS	Operating System
xi	PNG	Portable Network Graphics
xi	PR	Precision Recall

Chapter 1

Introduction

Glaucoma is a chronic eye disease that results from visual nerve damage caused by intraocular pressure in the eye. In the early stages of glaucoma, there are no symptoms of vision loss, but as it progresses, it may result in irreversible blindness. It is often associated with an accumulation of pressure within your eye. Glaucoma is common among families. It usually happens later in life. Diagnosis of glaucoma in the clinical environment includes intraocular pressure measurement, visual field testing, or examination of the optical disk of fundus images. The majority of people have no symptoms. If symptoms occur, it is usually around the end of the illness. The primary sign is generally a loss of lateral vision or peripheral vision. There's no way to prevent glaucoma. But if you find it early, you may reduce your chance of eye damage.

1.1 Description

Glaucoma is a chronic eye illness that results from optic nerve damage caused by high intraocular pressure. There are no symptoms of glaucoma in the early stages, but as the disease continues, it can lead to irreversible blindness. The majority of the time, it's due to an increase in intraocular pressure. In the clinical setting, glaucoma is diagnosed through intraocular pressure measurement, visual field tests, or fundus imaging inspection of the optic disc. As a result, a computer-aided diagnosis system could be employed to aid in the early detection of glaucoma, reducing computational complexity.

1.2 Problem Formulation

Glaucoma is a neurodegenerative eye disease caused by excessive intraocular pressure inside the retina. It is one of the leading causes of blindness around the globe. It can result in total blindness if not discovered early enough. As a result, it's become critical to investigate and find effective approaches to combat the disease's detrimental impacts. Ophthalmologists use a number of labor-intensive and time-consuming thorough retinal testing methods to diagnose glaucoma. A system that can function properly without the usage of specialized equipment or medical practitioners, as well as one that is less time-consuming, is needed.

1.3 Motivation

Glaucoma is one of the primary causes of blindness, according to the World Health Organization (WHO), affecting more than 60 million people worldwide and is expected to reach 80 million by 2020. It is predicted to have an impact on 12 million individuals in India or around one-fifth of the country's population. The disease's ever-increasing demand has posed a threat to people around the globe. High myopia, diabetes, eye surgery, and hypertension are all factors that contribute to this illness. The use of corticosteroids (such as eye drops, tablets, inhalers, and lotions) may reduce the risk of developing glaucoma. To reduce the risk of vision loss, we will put up a framework that can detect the disease early on.

1.4 Proposed Solution

We have developed a system that focuses on detecting glaucoma using four deep learning models. The models chosen are:

1. VGG19
2. VGG19+LSTM
3. Inception v3
4. Inception v3+LSTM

Every model is being worked with K-fold cross-validation and data augmentation to overcome the limitation of a small dataset. The features extracted are used to classify the input image and are then projected to be either glaucomatous or healthy. Finally, the values obtained for various performance evaluation parameters are compared.

1.5 Scope of the Project

This project aims to provide early detection of ocular disease such as Glaucoma. We understood the existing system for detection and classification and what are the drawbacks of this system. We are using the ACRIMA dataset that consists of 705 fundus images (396 glaucomatous and 309 normal images), out of which 632 images are for training, 73 are for testing, with a 90-10 split.

Chapter 2

Review of Literature

C. Sharmila and N. Shanthi [1] studied transfer learning for Early and Advanced Glaucoma Detection with Convolutional Neural Networks. This paper uses a machine learning method where a model developed for one task is reused because the place to begin for a model on a second task and is especially accustomed improve accuracy. The pre-trained model Inception V3 is used as transfer learning for glaucoma diagnosis. Transfer learning with both feature extraction perspective and also the fine-tuning procedure is employed. The weights of convolutional layers from the dataset ImageNet are fine-tuned for the automated glaucoma detection model. Adam is an optimization algorithm accustomed optimize network weights. During this research, loss function binary cross-entropy is utilized to live the performance of the model. Keras, TensorFlow libraries are employed for glaucoma diagnosis. Execution is performed using the net GPU access provided by Kaggle. The automated glaucoma diagnostic model has been trained and tested with the ORIGA dataset. The model anticipated two classes glaucoma and not glaucoma with an accuracy of 91.36%. The performance of glaucoma automated diagnostic system is analyzed using sensitivity, specificity, and accuracy parameters. 82.60% sensitivity and 95.30% specificity were achieved. It has been perceived that more accuracy is achieved with a minimum number of epochs. Since the dataset size is little, the present test accuracy results likely acceptable. A method is required to hold out the study using additional datasets shortly to boost the accuracy of the automated glaucoma detection algorithm.

Arkaja Saxena, Abhilasha Vyas, Lokesh Parashar, Upendra Singh [2] claim that glaucoma could be a vision-related condition of the human eye. This disease is taken into account as irreversible and causes damage to eyesight. It is a complex situation to cater to, and proper detection is critical. This problem is detected early and might be improved or can result in vision loss. This paper presents an architecture for correct glaucoma detection supported by deep learning using the convolutional neural network (CNN). The CNN offers a hierarchical image structure to tell them apart. The proposed work will be evaluated in six layers, each composed of tens of thousands of images. Then it'll make the utilization of the CNN technique for the classification of the patterns found in patients full of the disease. Within the presented mechanism, a dropout mechanism is additionally applied to reinforce the performance of the actual approach. The first goal is to search out the foremost similar patterns among the conventional human eye and also the infected glaucoma eye. During this case, the stalled mechanism is additionally wont to achieve adequate performance in detecting glaucoma. The datasets SCES and ORIGA are used for the examination. The analysis is conducted for the dataset and also the resulting values are 82.2% and 88.2% for the ORIGA and SCES datasets respectively. It's found that the 'ORIGA' dataset gives better results than 'SCES' for detecting both early & advanced glaucoma within the eye of the patient.

Ali Serener, Sertan Serte [3] reviewed transfer learning for early and advanced detection of glaucoma with convolutional neural networks. This paper uses deep convolutional neural networks to detect early and advanced stages of glaucoma on fundus images. The fundus images employed in this work have the three categories of no glaucoma (healthy), early glaucoma, and advanced glaucoma. The proposed model is performed on fundus images using two deep learning methods, a fifty-layer ResNet50, and GoogLeNet deep learning architecture for the classifications. The training for these models is completed employing a single NVIDIA GeForce GTX 1080Ti GPU running a Caffe deep learning framework. For best results, the training is first administrated on a unique dataset, so transfers the educational is employed to detect glaucomatous images of the fundus. This may reduce the disadvantage of getting fewer images for testing the dataset. ResNet-50 and GoogLeNet, are utilized to check the performance of those models on the RIM-ONE dataset. The performance of the 2 models is assessed in terms of precision, sensitivity, specificity, and surface under the ROC curve. The accuracy for ResNet50 classification is 0.86, the sensitivity is 0.21, the specificity is 0.93. However, the performance accuracy of the GoogLeNet model classification is 0.85, the sensitivity is 0.29, and also the specificity is 0.91. The results show that for early, advanced, and overall glaucoma detection GoogLeNet outperforms ResNet-50.

Sr. No	**Algorithm/ Year of issue**	**Advantages**	**Datasets**	**Limitations**	**Accuracy**	**Performance Evaluation Parameters**
1.	Inception V3 (2021)	Uses techniques to classify early	ORIGA	Not Mentioned	91.36	Sensitivity: 82.60% Specificity: 95.30%

		and advanced glaucoma.				
2.	Convolutional neural network (CNN) (2020)	Higher Detection capability, Achieved high accuracy.	SCES, ORIGA	Accuracy differs according to the dataset.	Accuracy between 82.2% to 88.2%	Sensitivity: 0.21 to 0.29 Specificity is 0.91 to 0.93
3.	GoogleLeNet and ResNet (2019)	Approximately one million images of ImageNet dataset are used to pre-train these models.	RIM-ONE	Not mentioned	85% to 86%	Sensitivity= 0.21 to 0.29 Specificity=0.91 to 0.93 ROC= 0.74 TO 0.75
4.	CNN + LSTM (2021)	Reduced glaucoma prediction time, High accuracy	RIMONE, DRISHTI-GS, DRIONS-DB	Cannot be trained without a labelled dataset.	Around 90%	Sensitivity= 95.4% Specificity= 96.7% AUC = 0.984
5.	MobileNet and Inception V3 (2021)	Improvement in Accuracy	ORIGA, SCES	A few cases that were misdiagnosed in terms of distributed colors and shapes	0.86 for MobileNet 0.90 for Inception V3	Precision= 0.87 to 0.90 Recall = 0.87 to 0.90 F1 score= 0.87 to 0.90 AUC = 83.1% to 88.7%
6.	Inception v3 (2021)	Reduced glaucoma prediction time, High accuracy	ACRIMA, LAG	Cannot be trained without a labelled dataset.	85.29%	Sensitivity= 95.4% Specificity= 96.7% AUC = 0.984
7.	AlexNet, SVM (2020)	Combining CNN and SVM resulted in greater accuracy than low-performance extraction methods.	HRF, Origa and Drishti_GS1	Not Mentioned	91.21%	Specificity:90.8%, Sensitivity: 85%
8.	VGG, ResNet	Uses forward	LAG	Not mentioned.	80 to 87%	Precision: 86.9%

	(2021)	and backward propagation to obtain the best weights with the lowest loss for the best performance			Best Accuracy: 86.9%.	Recall: 86.9%

9.	MNet and DeNet (2019)	Detects glaucoma on datasets having different brightness levels,More precise CDR	ORIGA,SCES	Accuracy and AUC can be improved.	Around 85%	AUC (0.8998 to 0.9183)
						Sensitivity(0.7609 to 0.8478)
						Specificity(0.8706 to 0.8380)

10.	VGG 16, VGG 19, Inception V3, ResNET 50, Xception (2019)	1.Uses huge amount of unlabeled data 2.The architecture of the classifier is less complex.	ACRIMA and Public databases	Performance decreased when testing the CNN on databases different from those used for training	90.17%	1.ROC curve(0.831) 2.AUC= 0.9605 3.Specificity (0.8580) 4. Sensitivity(0.9346) 5.F-score (0.8429)
11.	UNet, and EffcientNet (2022)	Higher Accuracy, Improved Efficiency	DRISHTI-GS, RIM-One	Accuracy differs according to the dataset.	Not mentioned	Dice Score = 0.96 Intersection Over Union (IOU) = 0.91
12.	AG-CNN (2019)	Used attention maps to improve the detection and pathological area localization.	LAG,RIM-ONE	1.Requires preprocessing and post-processing 2.Need to extract individual channels for segmentation.	85.2 % to 96.2%	Sensitivity= 84.8% to 95.4%,Specificity= 85.5% to 96.7%,AUC =0.916 to 0.984,F2 Score= 0.837to 0.954

Table 2.1: Literature Review

Chapter 3

System Analysis

3.1 Functional Requirements

- The system requires retinal fundus images of patients
- Feature selection, extraction, and classification will be done by the deep learning model
- The system should detect whether the patient has glaucoma or not

3.2 Non-functional Requirements

- Performance accuracy for the system application in the medical field, the result should be accurate for the complete dataset and the new images which will be added.
- The data of each patient must remain safe against illegal usage.
- The results obtained should be shown to the ophthalmologists.
- The output should be easy to view and simple to understand for ophthalmologists for further evaluation.

3.3 Specific Requirements

Hardware: The hardware environment consists of the following:

- CPU: Two Intel® Xeon® CPU E5-2650 v4 @2.20 GHz
- GPU: nVidia Titan Xp 12 GB GDDR5X
- Memory: 256 GB ECC DDR4 RAM (32 GB×8)
- Hard disk space: 20GB or more
- Display: Color Monitor

Software Development Tools:

- Python Editor
- Web Browser
- Operating System: Windows 10

3.4 Use-case Diagram

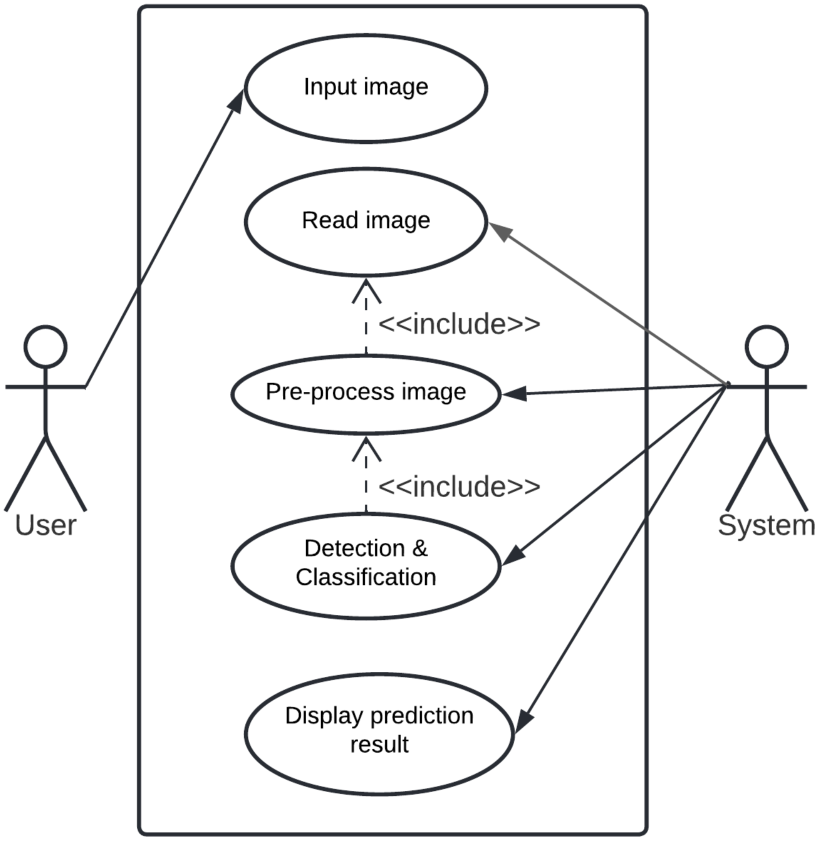


Figure 3.1: Use Case Diagram for Glaucoma Detection and Classification

Figure 3.1 above shows the use case diagram for Glaucoma Detection System. A USER is any person who wishes to check if they have Glaucoma. A SYSTEM is a trained model which accepts the retinal image as an input and gives the glaucoma detection as an output.

Table 3.1: Use case 1- Input image

Use case	Input image
Use Case ID	UC01
Actor	User
Description	User needs to Input a fundus image into the system.

Table 3.2: Use case 2- Read image

Use case	Read image
Use Case ID	UC02
Actor	System
Description	The system reads the fundus image that has been inputted by the user.

Table 3.3: Use case 3- Pre-process image

Use case	Pre-process image
Use Case ID	UC03
Actor	System
Description	The system performs pre-processing on the input image.

Table 3.4: Use case 4- Detection and Classification

Use case	Detection and Classification
Use Case ID	UC04
Actor	System
Description	The image is detected and classified

Table 3.5: Use case 5- Display Prediction Result

Use case	Display Prediction Result
Use Case ID	UC05
Actor	System
Description	The system displays the prediction (Glaucomatous or Normal)

Description: Tables 3.1 to 3.5 above show each use case along with their actor and description. As shown in Figure 3.1, the user gathers the image which will be used for detection. The system takes the image as input and performs pre-processing, followed by classification. Finally, the system predicts the image as Glaucomatous or Normal.

Chapter 4

Analysis Modeling

The tools and charts required to demonstrate the workflow of our project are documented as follows:

4.1 Flow chart

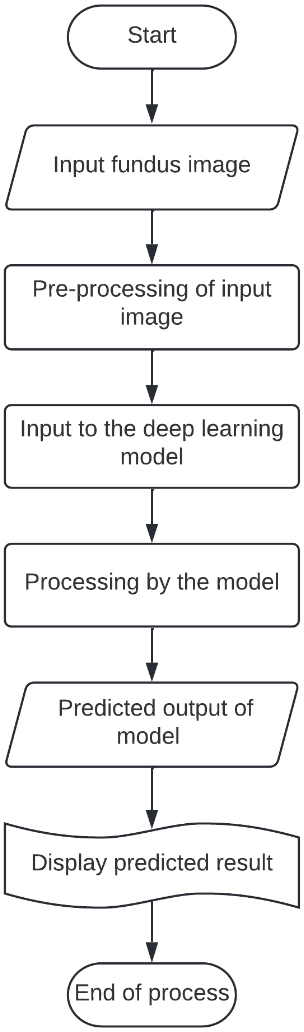

Start
Input fundus image
Pre-processing of input image
Input to the deep learning model
Processing by the model
Predicted output of model
Display predicted result
End of process

Figure 4.1: Flowchart defining the flow of processes of the System.

Description: As shown in the above figure 4.1, this flowchart defines the flow of the process that takes place in the procedure of detecting Glaucoma.

4.2 Activity Diagram

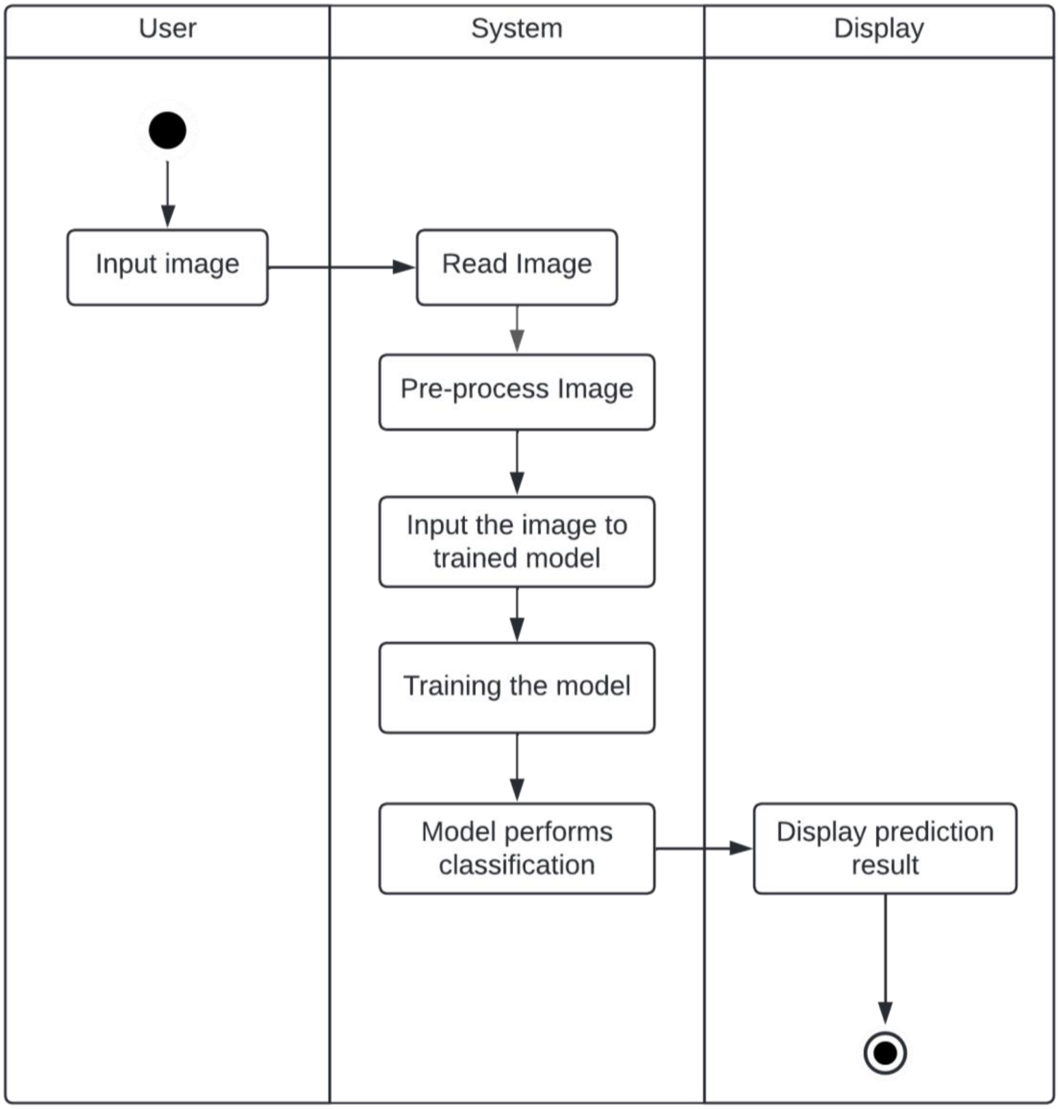

User
System
Display
Input image
Read Image
Pre-process Image
Input the image to trained model
Training the model
Model performs classification
Display prediction result

Figure 4.2: Activity Diagram for Glaucoma Detection and Classification

Description: As shown in the Figure 4.2, image is provided by the user and further steps such as image pre-processing, model training and classification is done by the system. The system then displays the predicted results.

4.3 Functional Modeling

4.3.1 DFD: Level 0

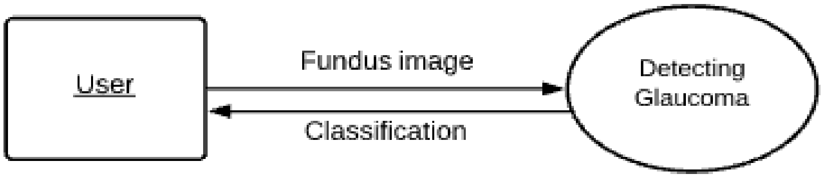


Figure 4.3.1: DFD Level 0

Description:

As shown in Figure 4.3.1, the user uploads the fundus image. The system then receives this image and processes it to display the detection to the end user.

4.3.2 DFD: Level 1

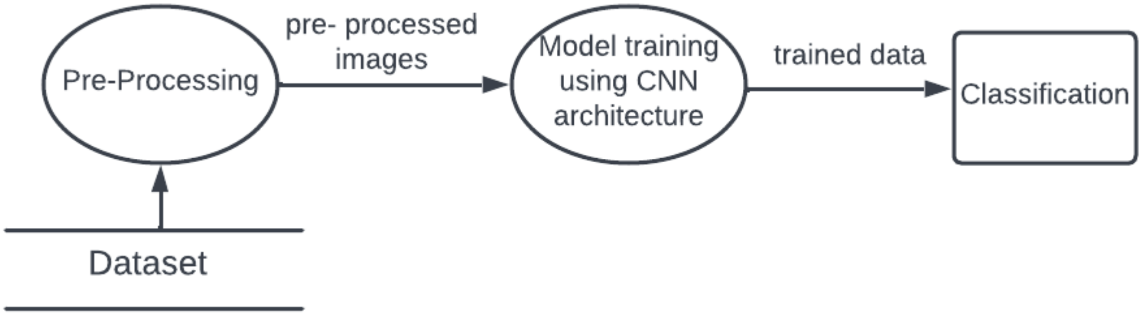


Figure 4.3.2: DFD Level 1

Description:

As shown in the Figure 4.3.2, the images collected undergoes processing using CNN architecture and the necessary results are provided to the ophthalmologist.

4.3.3 DFD: Level 2

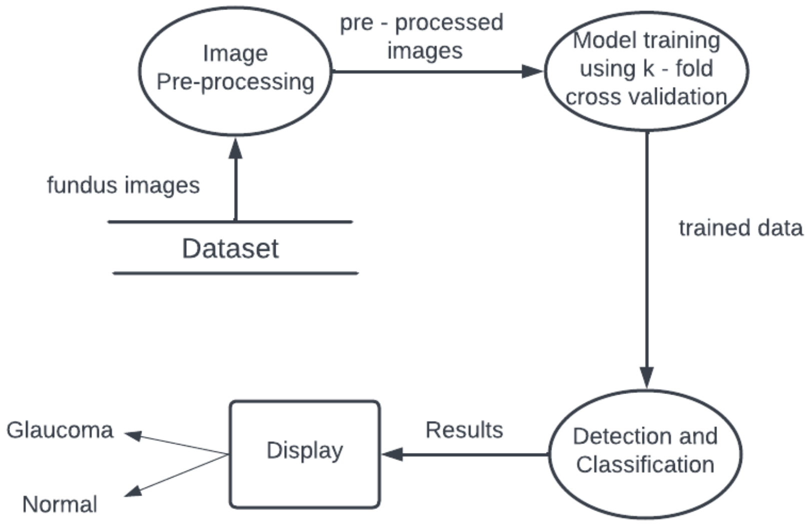
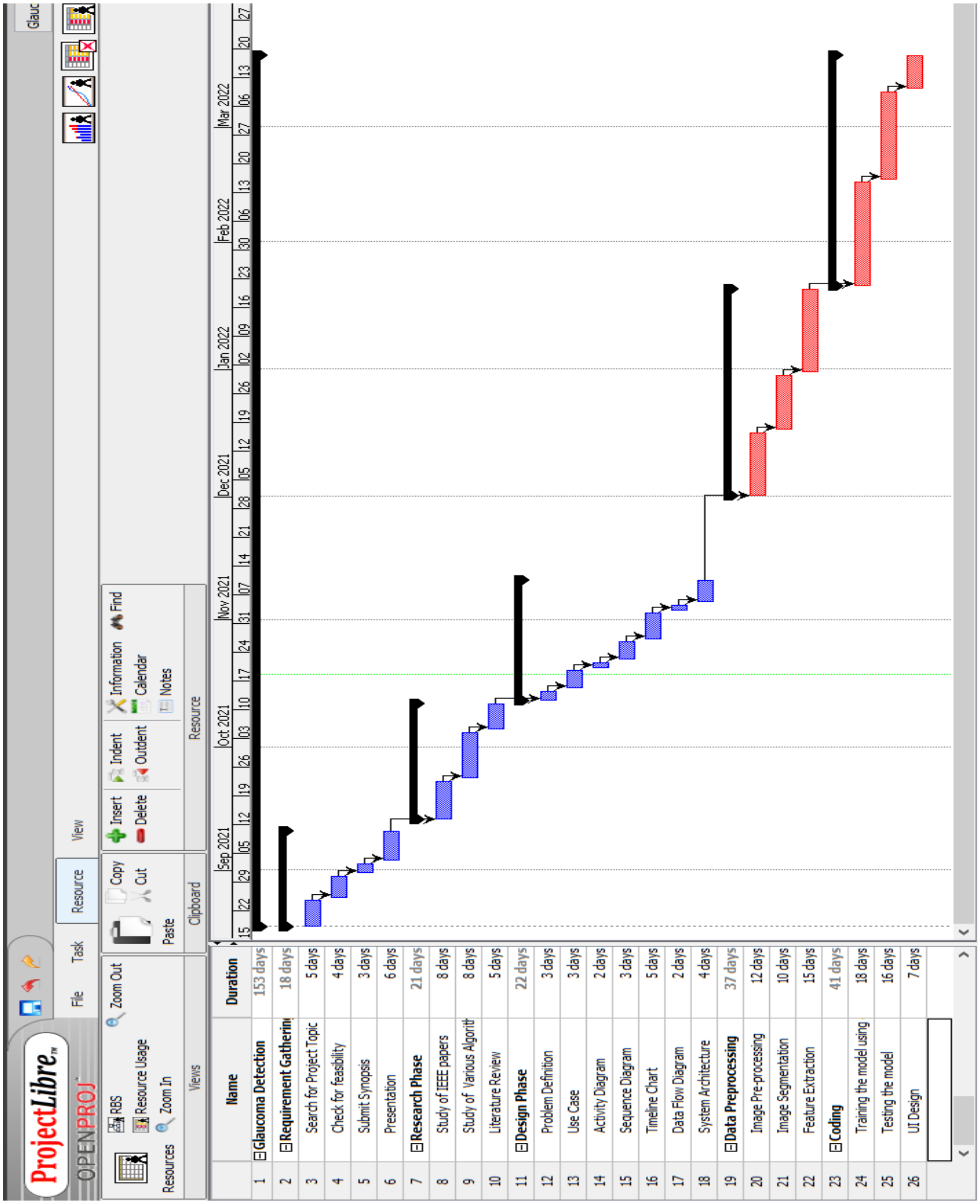


Figure 4.3.3: DFD Level 2

Description: As shown in the Figure 4.3.3, the images collected undergoes pre-processing and augmentation. Furthermore, the features are extracted and then the dataset is then modeled using CNN architecture. Once the classification (normal eye or glaucomatous eye) has been carried out, the results are then provided to the ophthalmologists.

4.4 Timeline Chart



	Name	Duration
1	⊟Glaucoma Detection	153 days
2	⊟Requirement Gatherin	18 days
3	Search for Project Topic	5 days
4	Check for feasibility	4 days
5	Submit Synopsis	3 days
6	Presentation	6 days
7	⊟Research Phase	21 days
8	Study of IEEE papers	8 days
9	Study of Various Algorit	8 days
10	Literature Review	5 days
11	⊟Design Phase	22 days
12	Problem Definition	3 days
13	Use Case	3 days
14	Activity Diagram	2 days
15	Sequence Diagram	3 days
16	Timeline Chart	5 days
17	Data Flow Diagram	2 days
18	System Architecture	4 days
19	⊟Data Preprocessing	37 days
20	Image Pre-processing	12 days
21	Image Segmentation	10 days
22	Feature Extraction	15 days
23	⊟Coding	41 days
24	Training the model using	18 days
25	Testing the model	16 days
26	UI Design	7 days

Figure 4.4 : Timeline chart

Chapter 5

Design

Here, we present the design aspect of our project such as the system architecture.

5.1 Workflow

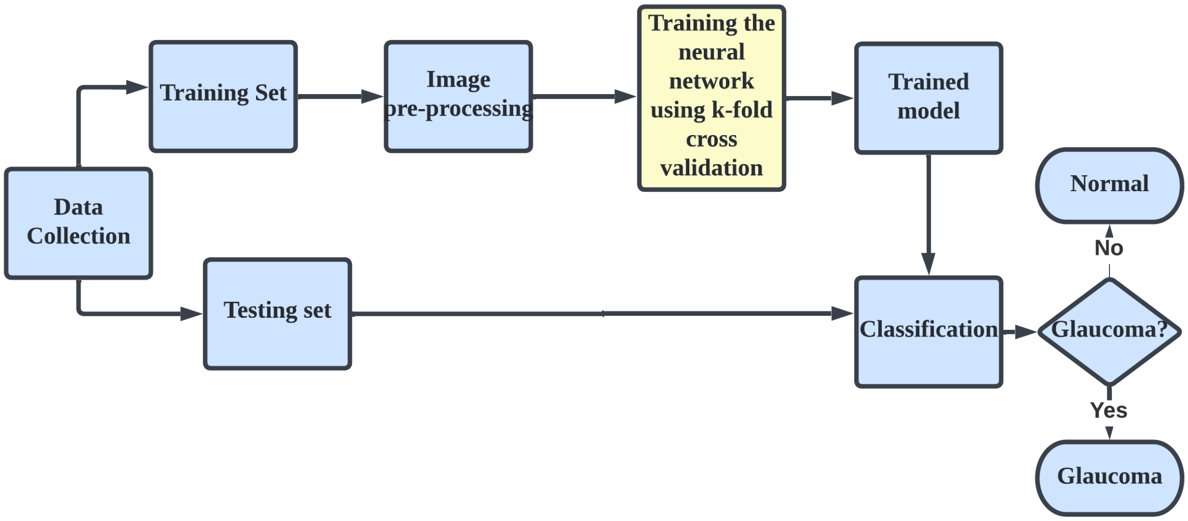


Figure 5.1 : Architecture of the System

Description: As seen in Figure 5.1, the data is collected and divided into Training set and Testing Set. The images from the training dataset further undergo Image pre-processing. Every model is being worked with K-fold cross-validation and data augmentation to overcome the limitation of a small dataset. The features extracted are used to classify the input image and are then projected to be either glaucomatous or normal.

Chapter 6

Implementation

6.1 Algorithms / Methods Used

In this work, we did a comparison study between four architectures: VGG-19, Inception v3, VGG-19 + LSTM and Inception v3 + LSTM . These architectures were chosen because they have a lot of experience with medical image categorization.

1. VGG- 19 Architecture:

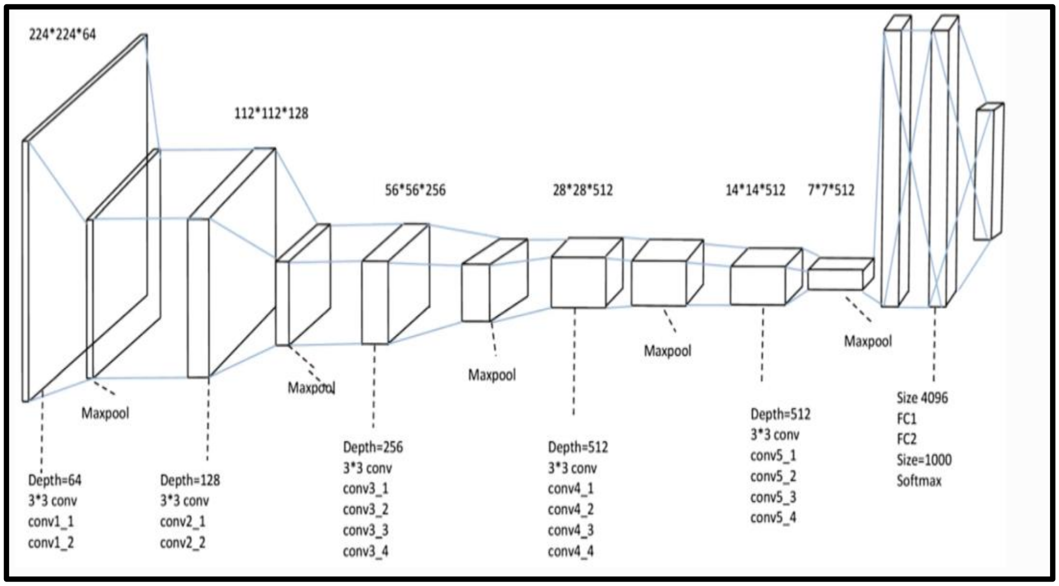


Figure 6.1 : VGG-19 Architecture [Dhillon, A., Verma, G.K, 2020][13]

VGG-19 is a convolution neural network that is 19 layers deep. It consists of 16 convolution layers, 5 MaxPool Layers, 3 Fully connected Layers and 1 sigmoid Layer. The first 16 layers have convolution and max pooling layers and are used for spatial feature extraction. After extracting all the features, the last 3 layers are used for image classification. A fixed image of size of (224*224) RGB image was given as input to this network which means that the matrix was of shape (224,224,3). For instance, 112*112*128, where 112*112 is the size and 128 is the number of filters, kernels. The filter size of the convolution layers are 3*3 and the stride is set to 1.The filter size of the max pooling layers are 2*2 and the stride is set to 2. After the final pooling layer, 7*7*512 volume is flattened into Fully connected (FC) layer with 4096 channels that is then followed by a sigmoid which classifies an image as Glaucomatous or Normal.

2. Inception v3:

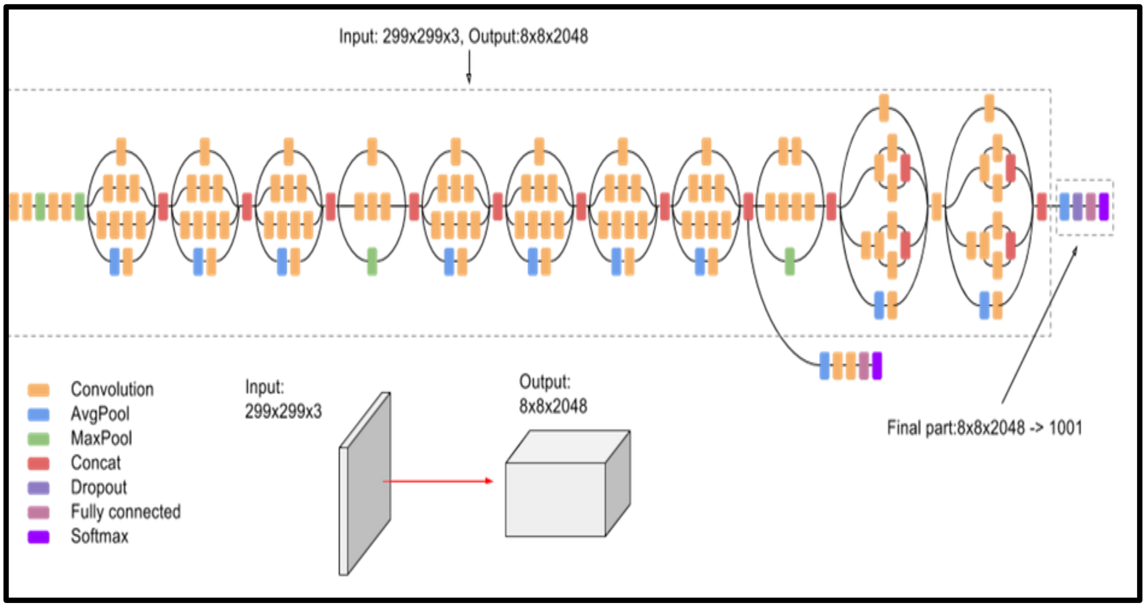


Figure 6.2 : Inception v3 Architecture [Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, Zbigniew Wojna, 2014][14]

Inception v3 mainly focuses on burning less computational power by modifying the previous Inception architectures.it has a total of 42 layers and a lower error rate than its predecessors.In comparison to VGGNet, Inception Networks (GoogLeNet/Inception v1) have proved to be more computationally efficient, both in terms of the number of parameters generated by the network and the economical cost incurred (memory and other resources)Inception v3 is a widely-used image recognition model that has been shown to attain greater than 78.1% accuracy on the ImageNet dataset. The model itself is made up of symmetric and asymmetric building blocks, including convolutions, average pooling, max pooling, concatenations, dropouts, and fully connected layers.

3. VGG- 19 + LSTM Architecture & Inception v3 + LSTM Architecture:

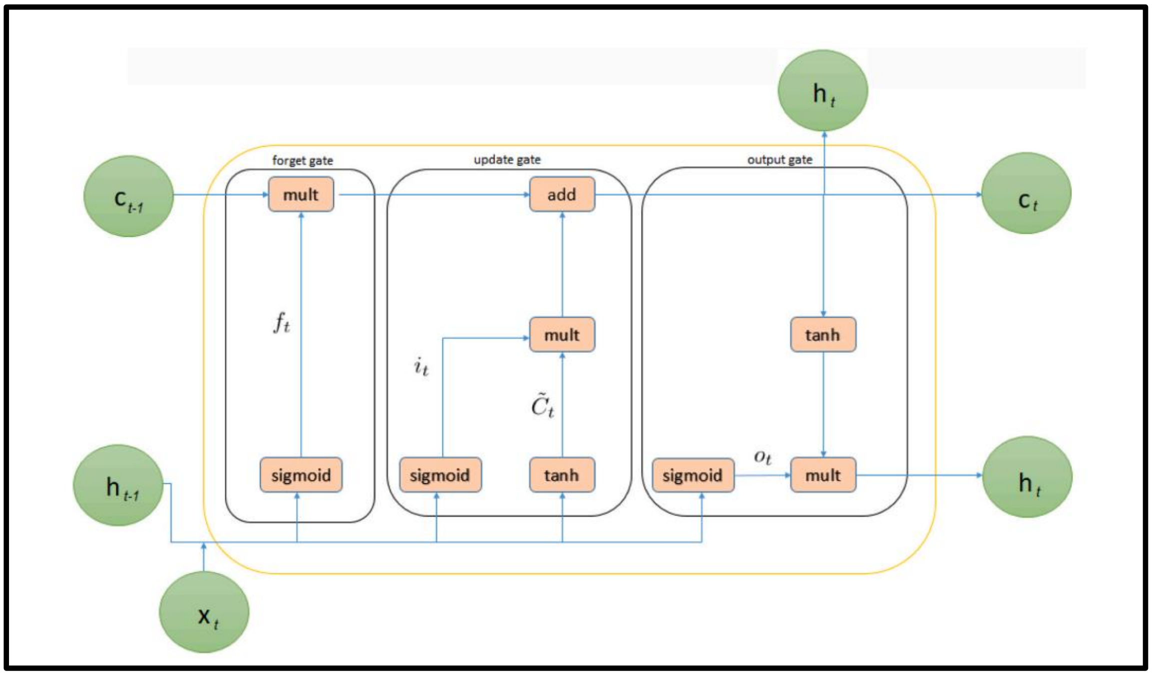


Figure 6.3: LSTM Architecture [Md. Zabirul Islam, Md. Milon Islam, Amanullah Asraf,2020][15]

To extract spatial and temporal data, we also created a mixed CNN (VGG-19 and Inception v3) and RNN (LSTM) architecture. A recurrent neural network is a type of artificial neural network that has loops that allow data to be stored inside the network. Recurrent neural networks utilize their reasoning from previous experiences to predict future occurrences. In sequences with spatial input, standard RNN architecture cannot be used directly. As a result, a more sophisticated architecture is required to complete jobs that need visual sequences to anticipate a result. As a result, we choose the CNN-RNN architecture.The CNN extracts spatial data from each video by converting it into consecutive pictures. The outputs are then used to detect temporal properties inside the image sequence using a recurrent sequence learning model (i.e. LSTM). Finally, the aggregated features are sent to a fully linked layer, which predicts categorization for the whole input sequence. The three gates added are: Forget Gate, Input Gate, and Output Gate. These gates help the LSTM cell to learn the long time dependency of a sequence and also helps to deal with vanishing gradient problems that occur in standard RNN.

6.2 Working of the project

6.2.1 Code

Code for VGG-19 + LSTM

Loading dataset paths:

```
train_path = '/content/drive/My Drive/GC/acrima/train'
test_path = '/content/drive/My Drive/GC/acrima/test'
external_path='/content/drive/My Drive/testsample'
#load dataset. Paths and targets
train_data = load_files(train_path)
train_filenames = np.array(train_data['filenames'])
train_targets = np_utils.to_categorical(np.array(train_data['target']))
```

Convert images to arrays:

```
def from_path_to_tensor(img_path):
    '''
    Read path of images and convert them to numpy arrays
    '''
    img = image.load_img(img_path, target_size=(224, 224))
    arr_im = image.img_to_array(img)
    return np.expand_dims(arr_im, axis=0)

x_train = np.vstack([from_path_to_tensor(im_path) for im_path in train_filenames])
x_test = np.vstack([from_path_to_tensor(im_path) for im_path in test_filenames])
x_external = np.vstack([from_path_to_tensor(im_path) for im_path in external_filenames])
print('x_train shape: {}'.format(x_train.shape))
print('x_test shape: {}'.format(x_test.shape))
print('x_external shape: {}'.format(x_external.shape))
```

Displaying samples of dataset:

```
import numpy as np
import matplotlib.pyplot as plt
fig= plt.figure(figsize = (16,8))
for i in np.arange(18):
    ax = fig.add_subplot(3, 6, i+1, xticks=[], yticks = [])
    ax.imshow(x_train[i].astype('uint8'))
```

Mean over all axes, i.e., of the entire 3D tensor:

```
mean = np.mean(x_train)
std = np.std(x_train)
```

Normalization. Factor 1e-7 added to avoid division by zero if std = 0:

```
x_train = (x_train - mean) / (std + 1e-7)
x_test = (x_test - mean) / (std + 1e-7)
x_external = (x_external - mean) / (std + 1e-7)
```

Create instance of class ImageDataGenerator:

```
datagen = ImageDataGenerator(rotation_range = 15,
                             width_shift_range = 15,
                             height_shift_range = 0.1,
                             horizontal_flip = True)

datagen.fit(x_train)
```

Show the effect of data augmentation on one image as an example:

```
i=0
x = x_train[10].reshape( (1,) + x_train[10].shape)

for batch in datagen.flow(x, batch_size =1):
    plt.figure(i)
    plt.imshow(image.array_to_img(batch[0]))
    i+=1
    if i % 4 == 0:
        break
plt.show()
```

Architecture code for VGG19+LSTM:

```
def build_model():
    conv_base = vgg19.VGG19(weights = 'imagenet', include_top= False, input_shape = (224,
224, 3))
    for layer in conv_base.layers[:-16]:
        layer.trainable = False

    #add the classifier layers
    last_output = conv_base.output
    x=Reshape((49,512))(last_output)
    x=LSTM(512,return_sequences=True)(x)
    x=LSTM(512,return_sequences=True)(x)
    x = Dropout(0.2)(x)
    x=Flatten()(x)
    x=Dense(128, activation="relu")(x)
    x=Dense(64, activation="relu")(x)
    x = Dense(1, activation='sigmoid', name = 'sigmoid')(x)
    model = Model(inputs=conv_base.input, outputs=x)
    model.compile(Adam(learning_rate=0.0001), loss='binary_crossentropy',
metrics=['accuracy'])
    return model
```

K-fold cross validation and Training phase:

```
k = 3
num_val_samples = x_train.shape[0] // k
num_epochs = 35
all_acc_scores = []
all_loss_scores = []

for i in range(k):
    print('processing fold #', i)
    x_val = x_train[i * num_val_samples: (i+1)*num_val_samples]
    y_val = y_train[i * num_val_samples: (i+1)*num_val_samples]

    x_partial_train = np.concatenate(
        [x_train[:i*num_val_samples],
         x_train[(i+1)*num_val_samples:]],
        axis = 0
    )

    y_partial_train = np.concatenate(
        [y_train[:i*num_val_samples],
         y_train[(i+1)*num_val_samples:]],
        axis = 0
    )

    model = build_model()
    checkpointer = ModelCheckpoint(filepath='kfold1vgg1900lstm.hdf5', save_best_only=True,
verbose = 1)
    early_stop = EarlyStopping(monitor='val_loss', patience=20)
    lambda_call= LambdaCallback(on_batch_end=lambda batch,logs:print(logs))

    partial_train_generator = datagen.flow(x_partial_train, y_partial_train, batch_size =
num_val_samples)

    history = model.fit(partial_train_generator, validation_data = (x_val, y_val), steps_per_epoch
= x_partial_train.shape[0] // num_val_samples,\
    epochs=num_epochs, verbose=1, callbacks=[checkpointer, lambda_call])
```

Classification report:

```
from sklearn.metrics import classification_report

print('Classification Report')
target_names = ['normal', 'glaucoma']
print(classification_report(np.argmax(y_test, axis=1),
                    np.argmax(y_prob_predic, axis=1),target_names = ['normal', 'glaucoma']))
```

Architecture code for VGG19:

```
def build_model():
    conv_base = vgg19.VGG19(weights = 'imagenet', include_top= False, input_shape = (224,
224, 3))
    for layer in conv_base.layers[:-16]:
        layer.trainable = False
    #add the classifier layers
    last_output = conv_base.output
    x=Flatten()(last_output)
    x=Dense(256, activation='relu')(x)
    x=Dense(64, activation='relu')(x)
    x = Dropout(0.2)(x)
    x = Dense(1, activation='sigmoid', name = 'sigmoid')(x)

    model = Model(inputs=conv_base.input, outputs=x)

    model.compile(Adam(learning_rate=0.0001), loss='binary_crossentropy',
metrics=['accuracy'])

    return model
```

Architecture code for Inception v3 + LSTM:

```
def build_model():
    conv_base = InceptionV3(weights = 'imagenet', include_top= False, input_shape = (299, 299,
3))
    for layer in conv_base.layers[:-16]:
        layer.trainable = False

    #add the classifier layers
    last_output = conv_base.output
    x=GlobalAveragePooling2D()(x)
    x = Dropout(0.25)(last_output)
    x=Flatten()(x)
    x = Dense(64, activation='relu')(x)
    x = Dense(1, activation='sigmoid', name = 'sigmoid')(x)
    model = Model(inputs=conv_base.input, outputs=x)
    model.compile(Adam(learning_rate=0.0001), loss='binary_crossentropy',
metrics=['accuracy'])
    return model
```

Architecture code for Inception v3:

```
def build_model():

    conv_base = InceptionV3(weights = 'imagenet', include_top= False, input_shape = (299, 299,
3))
      for layer in conv_base.layers[:-16]:
        layer.trainable = False

    #add the classifier layers
    last_output = conv_base.output
    x=Reshape((64,2048))(last_output)
    x=LSTM(2048, return_sequences=True)(x)
    x=LSTM(2048,return_sequences=True)(x)
    x=GlobalAveragePooling2D()(x)
    x = Dropout(0.25)(last_output)
    x=Flatten()(x)
    x = Dense(64, activation='relu')(x)
    x = Dense(1, activation='sigmoid', name = 'sigmoid')(x)
    model = Model(inputs=conv_base.input, outputs=x)
    model.compile(Adam(learning_rate=0.0001), loss='binary_crossentropy',
metrics=['accuracy'])
    return model
```

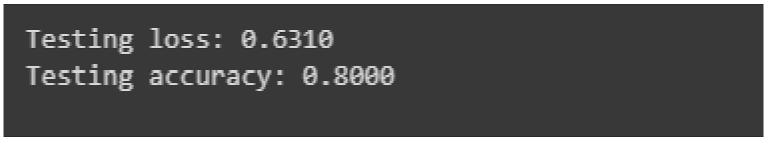
Chapter 7

Observation:

7.1 Experimental Setup:

The ACRIMA dataset is being used, and it consists of 705 fundus images (396 glaucomatous and 309 normal images).

632 images are being used for training, and 73 is being used for testing, with a 90-10 split.



```
Testing loss: 0.6310
Testing accuracy: 0.8000
```

Figure 7.3: Testing loss and accuracy

Figure 7.1.1: Glaucomatous Image

Figure 7.1.2: Normal Image

7.2 Performance evaluation parameters:

- Accuracy: It is the proportion of the systems correct predictions for the overall number of predictions. It determines how often the classifier will show the correct output. The ideal value is 1, and the worst case value is 0.

$$\text{Accuracy} = \frac{TP + TN}{TP + FN{+}TN{+}FP} \qquad (1)$$

- F1-Score: It is an indicator of the accuracy of a model on a dataset. It is used to test binary classification systems that classify examples as “positive” or “negative”. It is the harmonic mean of the precision and recall of the model. The ideal value is 1 ,and the worst case value is 0.

$$\text{F1} = \frac{2\ \times\ precision\ \times recall}{precision + recall} \qquad (2)$$

- Precision:It is the proportion of True Positives divided by the number of True Positive and False Positive. The ideal value is 1 ,and the worst case value is 0.

$$\text{Precision} = \frac{TP}{TP + FP} \qquad – (3)$$

- Recall: Recall is calculated as the number of true positives divided by the total number of true positives and false negatives. The ideal value is 1,and the worst case value is 0.

$$\text{Recall} = \frac{TP}{TP + FN} \quad – (4)$$

7.3 Type of Testing used

Black Box Testing

Is a software testing method in which the functionalities of software applications are tested without having knowledge of internal code structure, implementation details and internal paths.

Black Box Testing mainly focuses on input and output of software applications and it is entirely based on software requirements and specifications. It is also known as Behavioral Testing.

Figure 7.2 Black Box Testing

In this method, the tester selects a function and gives input value to examine its functionality, and checks whether the function is giving expected output or not. If the function produces correct output, then it is passed in testing, otherwise failed. The test team reports the result to the development team and then tests the next function. After completing testing of all functions if there are severe problems, then it is given back to the development team for correction.

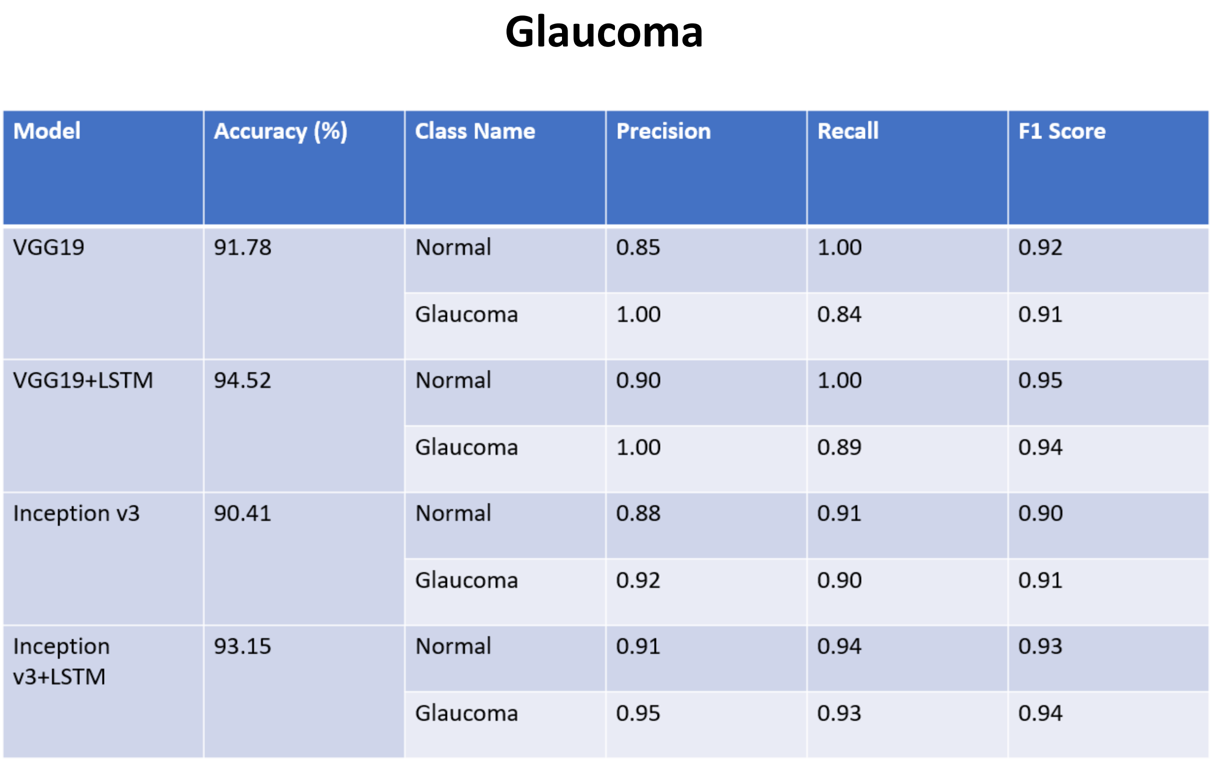
7.4 Test Cases

The results of the test cases have been summarized in the below table 7.1.

Test case ID	Test Case(Actions performed)	Expected result	Actual Outcome
1	Sample from ACRIMA dataset	Normal	Normal
2	Sample from RIM-ONE dataset	Normal	Glaucomatous
3	Sample from ACRIMA dataset	Glaucomatous	Glaucomatous
4	Sample from ACRIMA dataset	Glaucomatous	Glaucomatous
5	Sample from RIM-ONE dataset	Glaucomatous	Glaucomatous

Chapter 8

Results and Discussions



Glaucoma

Model	Accuracy (%)	Class Name	Precision	Recall	F1 Score
VGG19	91.78	Normal	0.85	1.00	0.92
		Glaucoma	1.00	0.84	0.91
VGG19+LSTM	94.52	Normal	0.90	1.00	0.95
		Glaucoma	1.00	0.89	0.94
Inception v3	90.41	Normal	0.88	0.91	0.90
		Glaucoma	0.92	0.90	0.91
Inception v3+LSTM	93.15	Normal	0.91	0.94	0.93
		Glaucoma	0.95	0.93	0.94

Table 8.1: Summary of Results using evaluation parameters

Observation for VGG19 model: The accuracy achieved is 91.78% with a better performance on normal images as compared to glaucoma images.

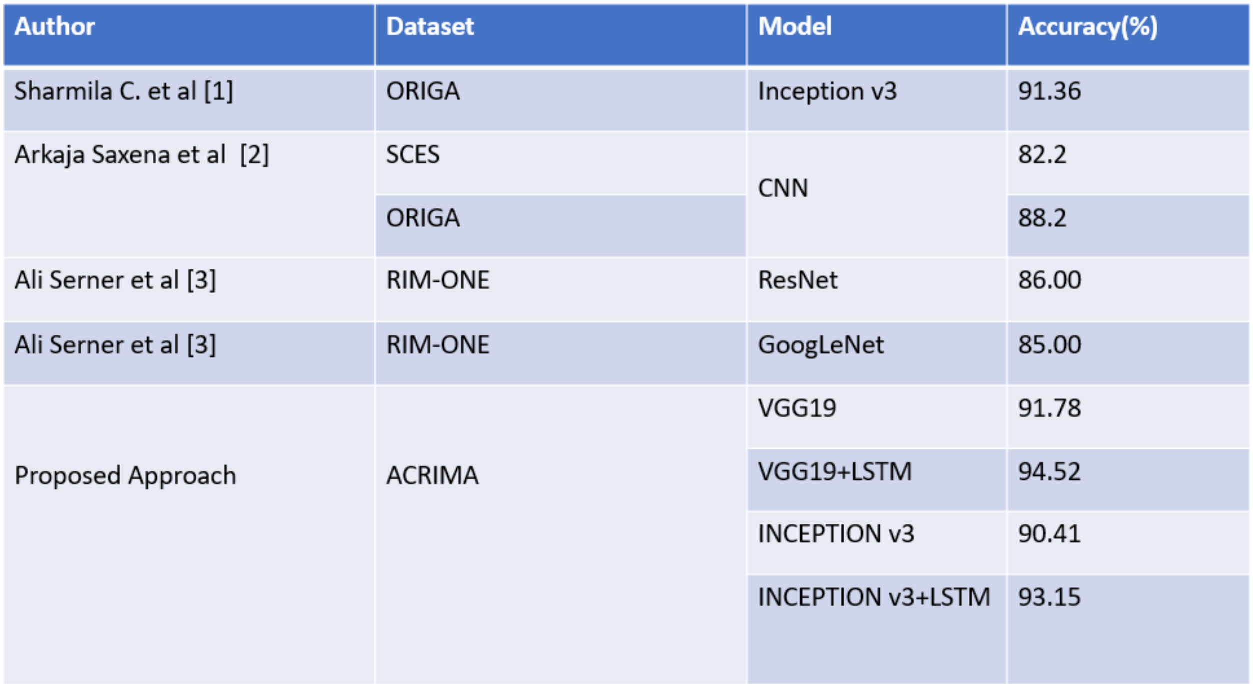
Observation for VGG19+LSTM model: The accuracy achieved is 94.52% with a better performance on normal images as compared to glaucoma images.

Observation for Inception v3 model: The accuracy achieved is 90.41% with a better performance on glaucoma images as compared to normal images.

Observation for Inception v3+LSTM model: The accuracy achieved is 93.15% with a better performance on glaucoma images as compared to normal images.

Therefore, it is noticeable that VGG19 based models perform better on normal images and Inception v3 based models perform better on glaucoma images.

Each row in the table corresponds to various approaches that we have used during the training. There were 4 models used for the training purpose. The k-fold cross validation technique was used for training purpose with k=3. This was done in order to overcome the limitation of a small dataset. After training the model for 35 epochs, the average training accuracy achieved was 96% for VGG19 based models, and the average testing accuracy was 93.15%. After training the model for 35 epochs, the average training accuracy achieved was 99% for Inception v3 based models, and the average testing accuracy was 91.78%. It is observed that the addition of LSTM to VGG-19 and Inception v3, helped to increase the training as well as testing accuracy. Moreover, we observed that the VGG 19- LSTM model produced best results with highest accuracy.



Author	Dataset	Model	Accuracy(%)
Sharmila C. et al [1]	ORIGA	Inception v3	91.36
Arkaja Saxena et al [2]	SCES	CNN	82.2
	ORIGA		88.2
Ali Serner et al [3]	RIM-ONE	ResNet	86.00
Ali Serner et al [3]	RIM-ONE	GoogLeNet	85.00
Proposed Approach	ACRIMA	VGG19	91.78
		VGG19+LSTM	94.52
		INCEPTION v3	90.41
		INCEPTION v3+LSTM	93.15

Table 8.2: Comparison study between various approaches

Precision-Recall Curve

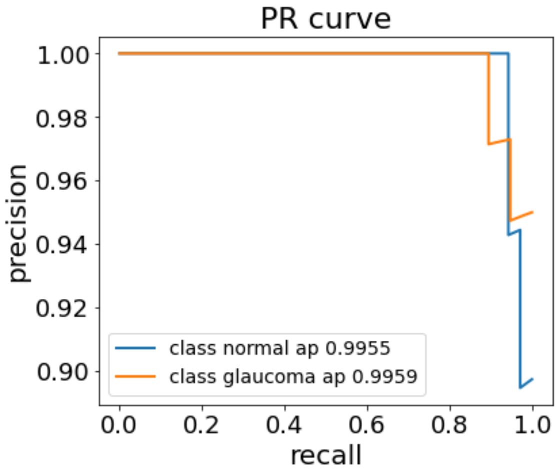


Fig 8.1: PR curve for VGG 19 model

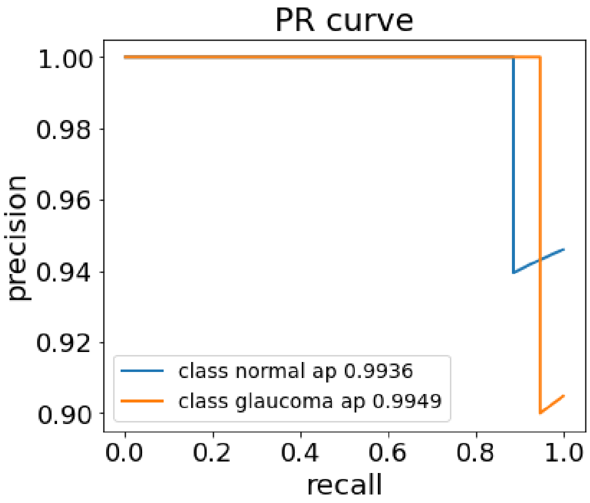


Fig 8.2: PR curve for VGG 19-LSTM model

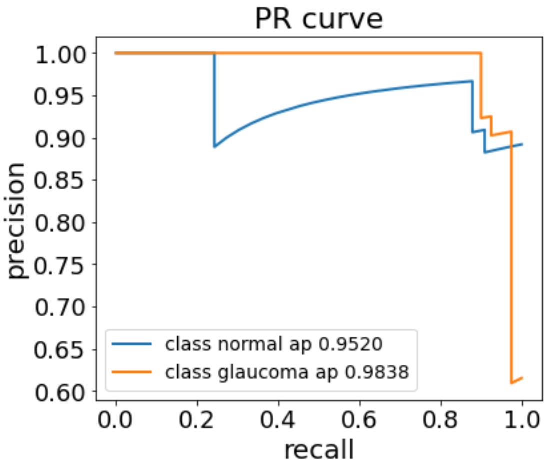


Fig 8.3: PR curve for Inception v3 model

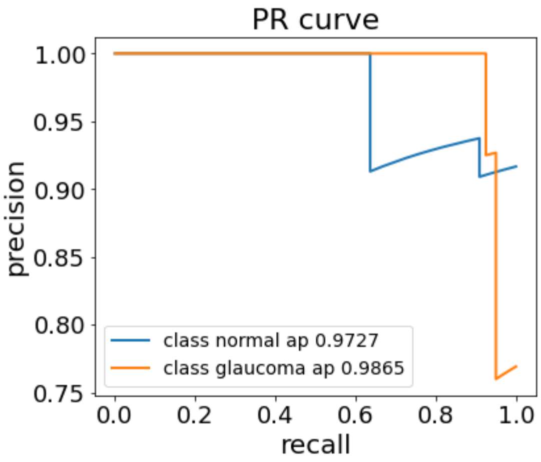


Fig 8.4: PR curve for Inception v3-LSTM model

Chapter 9

Conclusion and Future Scope

9.1 Conclusion

Globally, every year numerous people are affected by a retinal disease, which is known as glaucoma. It is one of the primary concerns of the human eye. If not detected early, it causes permanent blindness. So to detect this disease, a mechanism that uses CNN and RNN is used for the analysis. We have used various models, such as VGG19, VGG19+LSTM, Inceptionv3, Inceptionv3+LSTM, and after performing a comparative study on the ACRIMA dataset, it is noticeable that VGG19+LSTM is the best performing model. The VGG19 + LSTM model predicted two classes (glaucoma and non-glaucoma) with an accuracy of 94.52%. We have developed this deep learning model which has been successful in achieving state of the art accuracy in image classification which is comparable to a clinical setup. The aim of this project is to provide an effective, fast, and efficient solution for exposure of glaucoma in the eye.

These accurate results were obtained due to transfer learning and data augmentation. Data augmentation helped us tackle the problems while training CNNs. This initiative has the potential to become much bigger. Therefore, it is noticeable that VGG19 based models perform better on normal images and Inception v3 based models perform better on glaucoma images.

9.2 Future Scope

A model like this will be beneficial to both ophthalmologists and patients. The potential for future application of this concept is boundless, and it must be studied and pushed to new heights to uncover everything that can be accomplished. We can try to collect real- life data and wider datasets from hospitals all over India in order to study any changes in the retina due to different geographical locations. Also, we can work on categories based on the severity of Glaucoma.

References

[1] Sharmila C. & Shanthi N.. (2021). Retinal Image Analysis for Glaucoma Detection Using Transfer Learning. 10.1007/978-981-15-9019-1_21.

[2] A. Saxena, A. Vyas, L. Parashar and U. Singh, "A Glaucoma Detection using Convolutional Neural Network," 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), 2020, pp. 815-820, doi: 10.1109/ICESC48915.2020.9155930.

[3] A. Serener and S. Serte, "Transfer Learning for Early and Advanced Glaucoma Detection with Convolutional Neural Networks," 2019 Medical Technologies Congress (TIPTEKNO), 2019, pp. 1-4, doi: 10.1109/TIPTEKNO.2019.8894965.

[4]Demir F, Taşcı B. An Effective and Robust Approach Based on R-CNN+LSTM Model and NCAR Feature Selection for Ophthalmological Disease Detection from Fundus Images. J Pers Med. 2021 Dec 2;11(12):1276. doi: 10.3390/jpm11121276. PMID: 34945747; PMCID: PMC8709012.

[5] Olivas, L.G., Alférez, G.H. & Castillo, J. Glaucoma detection in Latino population through OCT's RNFL thickness map using transfer learning. Int Ophthalmol 41, 3727-3741 (2021). https://doi.org/10.1007/s10792-021-01931-w.

[6] Tasnim Afroze, Shumia Akther, Mohammed Armanuzzaman Chowdhury, Emam Hossain, Mohammad Shahadat Hossain, and Karl Andersson. (2021). Glaucoma Detection Using Inception Convolutional Neural Network V3. Springer International Publishing. Cham. DOI: 10.1007/978-3-030-82269-9_2

[7] Ajitha S, Dr. M V Judy, Dr. Meera N, Dr. Rohith N. (2020). Automated Identification of Glaucoma from Fundus Images using Deep learning Techniques. European Journal of Molecular & Clinical Medicine. Volume 07, Issue 02, 2020.

[8]A. Sallam et al., "Early Detection of Glaucoma using Transfer Learning from Pre-trained CNN Models," 2021 International Conference of Technology, Science and Administration (ICTSzA), 2021, pp. 1-5, doi: 10.1109/ICTSA52017.2021.9406522.

[9]Fu, Huazhu & Cheng, Jun & Xu, Yanwu & Liu, Jiang. (2019). Glaucoma Detection Based on Deep Learning Network in Fundus Image. 10.1007/978-3-030-13969-8_6.

[10]Diaz-Pinto, A., Morales, S., Naranjo, V. et al. CNNs for automatic glaucoma assessment using fundus images: an extensive validation. BioMed Eng OnLine 18, 29 (2019). https://doi.org/10.1186/s12938-019-0649-y.

[11] Garg, H., Gupta, N., Agrawal, R. et al. A real time cloud-based framework for glaucoma screening using EfficientNet. Multimed Tools Appl (2022). https://doi.org/10.1007/s11042-021-11559-8

[12]Xu, Mai & Liu, Hanruo & Li, Yang & Wang, Xiaofei & Jiang, Lai & Wang, Zulin & Fan, Xiang & Wang, Ningli. (2019). A Large-Scale Database and a CNN Model for Attention-Based Glaucoma Detection. IEEE transactions on medical imaging. PP. 10.1109/TMI.2019.2927226.

[13]Dhillon, A., Verma, G.K. Convolutional neural network: a review of models, methodologies and applications to object detection. *Prog Artif Intell* 9, 85–112 (2020). https://doi.org/10.1007/s13748-019-00203-0

[14]Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, Zbigniew Wojna .Rethinking the Inception Architecture for Computer Vision https://doi.org/10.48550/arXiv.1512.00567

[15]A Combined Deep CNN-LSTM Network for the Detection of Novel Coronavirus (COVID-19) Using X-ray ImagesMd. Zabirul Islam, Md. Milon Islam, Amanullah AsrafmedRxiv 2020.06.18.20134718; doi: https://doi.org/10.1101/2020.06.18.20134718

Acknowledgments

We hereby take the privilege to present our project report on Glaucoma Detection and Classification. We are very grateful to our Project Supervisor and Head of Department, Dr. Kavita Sonawane for contributing her valuable moments in our Project from her busy and hectic schedule right from the project's inception. Being after this like a true mentor, she has been a constant source of inspiration to us. Her suggestions have greatly contributed to the betterment of our project.

Our special thanks to staff members and lab assistants for their cooperation