A

Major Project Report

On

AUTOMATIC DETECTION AND CLASSIFICATION OF EYE CATARACT IN FUNDUS IMAGES USING DEEP LEARNING TECHNIQUE

(Submitted in partial fulfilment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

R. BHARGAVA CHARY (187R1A05A9)

S. PRATHYUSHA (187R1A05B6)

CH. NAGASAI TEJA (187R1A0568)

G. SHIRISHA (18C21A0553)

Under the Guidance of

N. BHASKAR

(Associate Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi)

Recognized Under Section 2(f) & 12(B) of the UGCAct.1956.

Kandlakoya(V), Medchal Road, Hyderabad-501401

2018-22

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled "**AUTOMATIC DETECTION AND CLASSIFICATION OF EYE CATARACT IN FUNDUS IMAGES USING DEEP LEARNING**" being submitted by **R. BHARGAVA CHARY (187R1A05A9), S. PRATHYUSHA (187R1A05B6), CH. NAGASAI TEJA (187R1A0568), G. SHIRIS(18C21A0553)** in partial fulfilment of the requirements for the award of the degree of B. Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by him/her under our guidance and supervision during the year 2021-22.

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

Dr. N. Bhasker
Associate Professor
INTERNAL GUIDE

Dr. A. Raji Reddy
DIRECTOR

Dr. K. Srujan Raju
HOD

EXTERNAL EXAMINER

Submitted for viva voice Examination held on ___________________

ACKNOWLEDGMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard to my guide **Mr. N. Bhaskar**, Associate Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to Project Review Committee (PRC) **Dr. M. Varaprasad Rao, Mr. J. Narasimha Rao, Dr. T. S. Mastan Rao, Dr. Suwarna Gothane, Mr. A. Uday Kiran, Mr. A. Kiran Kumar, Mrs. G. Latha** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Srujan Raju**, Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy**, Director for being cooperative throughout the course of this project. We also express our sincere gratitude to Sri. **Ch. Gopal Reddy**, Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

R. BHARGAVA CHARY (187R1A05A9)

S. PRATHYUSHA (187R1A05B6)

CH. NAGASAI TEJA (187R1A0568)

G. SHIRISHA (18C21A0553)

ABSTRACT

Cataract is the clouding of the lens in the eye which is painless and developed gradually over a long period. Cataract is a eye disorder which occurs when some of protein at lens clumped together that makes it dull and increase opacity of the lens, causing some loss of vision. Most of the cataract is related to aging. In this work we have presented VGG-19 Algorithm, the deep neural network for automatically detecting cataract in the fundus images. We used the images. We used the Ocular Disease Recognition Dataset (ODRD) , which consists of 5000 fundus images acquired with three different fundus cameras We have applied a VGG-19 architecture and trained the dataset ODRD and and classified the results where we had achieved Accuracy of 96% and loss of 0.993.

LIST OF FIGURES/TABLES

LIST OF SCREENSHOTS

TABLE OF CONTENT

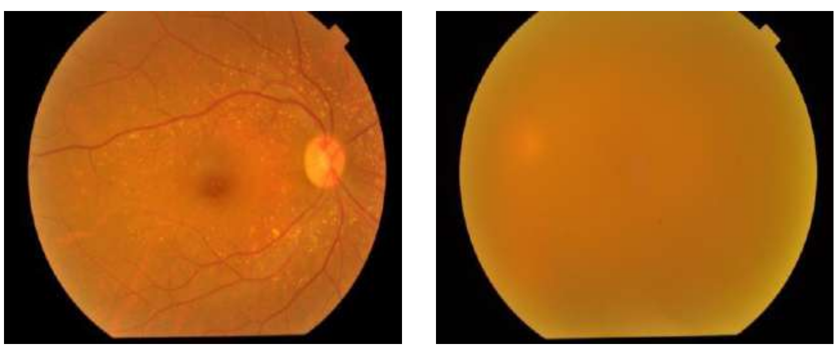
1.INTRODUCTION

1.INTRODUCTION

1.1 PROJECT SCOPE

We focus to develop a model that would detect cataract at early juncture such That future complications can be avoided. Rather than using a fundas image, which gives the rear view of eye, we have decided to use the from view of the eye here which involves several advantages such as low cost, easy to understand the usage, can easily reach common people who have no knowledge in the medical field.

Eyes are the unique vision gland in the human body and many people are suffering from eye disorder causing vision impairments. Cataract is one of the most prevalent eye diseases which is the frequent reason for blindness. Fig. 1 shows a retinal fundus image which represents the normal retinal fundus image that visible the capillaries and vascular cell (Fig. 1a). Fig. 1b shows a cataract image in which capillaries and vascular are not visible due to blurriness. At this stage, vision is lost in most of the people.



(a) (b)

Fig.1.1. Retinal Fundus Image (a) non-cataract (b) cataract

According to the World Health Organization (WHO), at least 2.2 billion people around the world are affected by blindness and vision impairments, around 1 billion people have the preventable vision impairments, more than 50% of the blindness is caused by cataract and around 40 million people will be blind by the year 2025 . Early detection and diagnosis may degrade the lamenting of cataract in cataract complaints and prevent visual impairment from turning into blindness.

1.2 PROJECT PURPOSE

Cataracts develop as a result of aging and the use of crystalline lenses. Many interdependent elements, including the lens' microscopic structure and chemical content, preserve the lens' transparency and optical homogeneity. A progressive deposit occurs in the lens where a yellow-brown pigment is seen which increases with aging. This also reduces the transmission of light into the eyes. The symptoms of cataract basically depend on the types of cataracts, the lifestyle of a person, and also his visual requirements. Intracapsular and extracapsular cataract extractions are the two terms used interchangeably. Intracapsular extraction entails removing the entire lens while keeping the capsule intact.

In the developed world, this approach is hardly used for treatment. It is still popular in underdeveloped countries since it requires fewer expensive and sophisticated instruments. It does not need a highly stable electricity supply. Besides that, it can be performed within a short training period. Another method is extracapsular extraction. The nucleus of the lens is removed in one piece; a relatively large incision is required. Cataract disease can be detected using transfer learning-based intelligent methods and ocular image datasets. Preliminary cataract detection and prevention may help to minimize the rate of blindness. This approach is cheap and efficient, which is the main motivation of this study.

In recent years, better cataract surgery has been created than in the previous 20 years. Around 85-90% of patients who experience cataract surgery will have 6/12 (20/40 or 0.5) best-corrected vision in patients with no ocular comorbidity such as macular degeneration, diabetic retinopathy, or glaucoma.We aim to detect cataract in an efficient way. In the sense, to make cataract detection with less intricacy and much lucid user interaction

1.3 PROJECT FEATURES

Artificial intelligence-based systems for cataract detection are mostly based on global features (e.g. discrete cosine transformation (DCT)), local features such as(e.g.,local standard deviation) and deep features (e.g., deep CNN) based methods have achieved higher accuracy. Although numerous deep-learning-based automatic cataract detection systems have been reported in the literature, they still suffer from limitations such as the low detection accuracy, a high number of model parameters, and thus computationally expensive.

A fundus image-based automatic cataract detection system is proposed in this to overcome the limitations mentioned above, which classifies the patients into two groups as the cataract or non-cataract conditions.The novelties of the proposed method are as follows: 1) reducing the number of parameters in the model such as layers and weights, thus reducing the computational cost and time,2) increasing the detection accuracy based on the proposed deep neural network structure. Thus, the proposed cataract detection can be used for a mass screening and cataract grading.

Our aim is to provide an effective solution to the cataract problem by giving optimized and cheaper alternatives to the existing methods of detection. As earlier detection of cataract will givethe higher rate of surgical success and our model will allow for better and more accurate diagnosis of the cataract condition.

2.SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

SYSTEM ANALYSIS

System Analysis is the most important phase in the system development process. The System is studied to the minute details and analyzed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In an analysis, a detailed study of these operations performed by the system and of their relationships within and outside the system is done. A key question considered here is, "what must be done to the solve the problem?" The system is viewed as a whole and the inputs to the system relational identified. Once analysis is completed the analyst has a firm understanding of what is to be become.

2.1 PROBLEM DEFINITION

In this model, the VGG19, a pre-trained model is applied as on Convolutional Neural Network (CNN) architecture which is deployed over the data set obtained. On modelling the system with collected samples, the neural network algorithm has resulted with an accuracy rate of 96.1% for cataract detection and finally the system Is trained to classify an input eye image as cataract or normal eye.

An intelligent digital eye cataract detection system would be the very useful and affordable for those people. Therefore, the develop of a intelligent eye cataract cataract detection system has been a research focus for some time. In recent years, many electronics devices have developed for a retinal fundus image capturing and detecting cataract but these devices are reported to be as expensive and have poor ability of detecting cataract. As a result, there is the need for the development of a low-cost computer assisted intelligent cataract detection system that will reduce the dependency on ophthalmologists or similar eye specialist for eye treatment such as the cataract detection. This paper proposed an automatic cataract detection system based on DCNNs that detect a cataract and non cataract from retinal fundus images. A trained classifier model based on VGG-19 is used for the cataract detection.

2.2 EXISTING SYSTEM

Artificial intelligence-based systems for cataract detection are mostly based on global features, local features and deep features (e.g., deep CNN which have achieved higher accuracy).

Although numerous deep learning-based automatic cataract detection systems have been reported in the literature, they still suffer from limitations such as low detection accuracy, a high number of model parameters, and thus being computationally expensive.

Res-Net 50 is a model which has achieved 94% of accuracy and it has high number of model parameters and being computationally expensive.

2.2.1 LIMITATIONS OF EXISTING SYSTEM

- The techniques used in existing system are not able to produce accurate values.
- It is complex to identify the cataract.
- It is an expensive procedure.

To avoid all these limitations and make the working more accurately the

more system needs to be implemented efficiently.

2.3 PROPOSED SYSTEM

We presented an automated cataract detection system namely VGG-19 based on deep learning and the dataset is rearranged, preprocessed and augmented to improve the dataset to feed deep network.

It classifies the patients into two groups:

Cataract or Non-Cataract conditions.

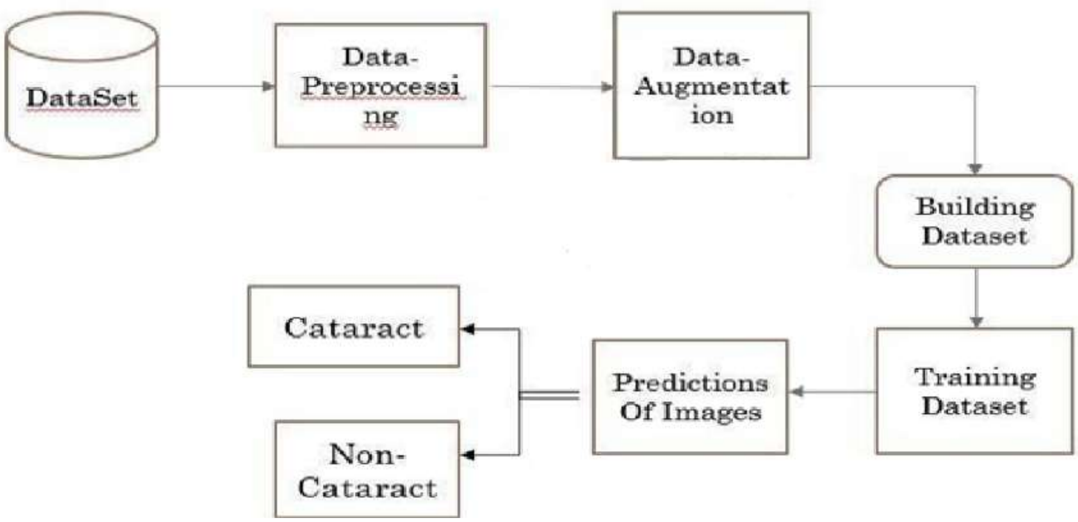


Figure 2.1 Proposed system

Through this we have extracted the cataract and normal information from the Dataset and Created a Dataset from Images.

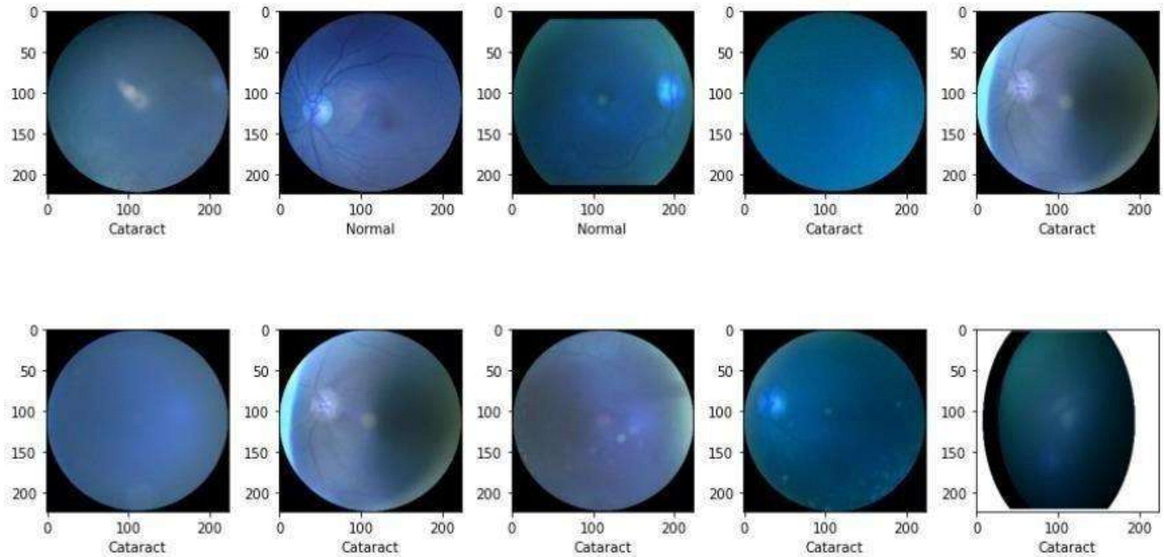


Figure 2.2 Input Data

2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM

- It introduces the possibility of using techniques to produce more accuracy.
- Increasing the detection accuracy based on the proposed deep Neural network structure.
- Reducing the number of parameters in the model such as layers And weights, thus reducing the computational cost and time
- These models can easily be adopted for other eye diseases within a short period of time.
- These models can also be extended to any disease that involves images.

2.4 FEASIBILITY STUDY

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

Three key considerations involved in the feasibility analysis are

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

2.4.1 ECONOMIC FEASIBILITY

The developing system must be justified by cost and benefit. Criteria to ensure that the effort is concentrated on a project, which will give best, return at the earliest. One of the factors, which affect the development of a system, is the cost it would require.

The following are some of the important financial questions asked during

Investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to the spend for the proposed system. Also all the resources are already available, it give in an most indication of the system is economically possible for development.

2.4.2 TECHNICAL FEASIBILITY

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

2.4.3 BEHAVIORAL FEASIBILITY

This includes the following questions :

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

2.5 HARDWARE & SOFTWARE REQUIREMENTS

2.5.1 HARDWARE REQUIREMENTS :

Hardware Requirements specifies the logical characteristics of each interface between the software product and the hardware components are of the system. The following are some hardware requirements.

- **RAM** : 8GB and Higher
- **Processor** : Intel i5
- **Hard Disk** : 500GB Minimum
- **Speed** : 2.4 GHz
- **Keyboard** : Standard Keyboard

2.5.2 SOFTWARE REQUIREMENTS :

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements.

Functional requirements for a secure cloud storage service are straightforward:

1. The service should be able to store the user‘s data;
2. The data should be accessible through any devices connected to the Internet;
3. The service should be capable to synchronize the user‘s data between multiple devices (notebooks, smart phones, etc.);
4. The service should preserve all historical changes (versioning);
5. Data should be shareable with other users;

- Operating System : Windows 10
- Python IDE : Python 3.9.5
- Tool : Anaconda(Jupyter notebook, spyder)

3. ARCHITECTURE

3.ARCHITECTURE

3.1 PROJECT ARCHITECTURE

The project architecture shows the procedure followed for automatic detection and classification of eye cataract in fundus images using deep learning technique.

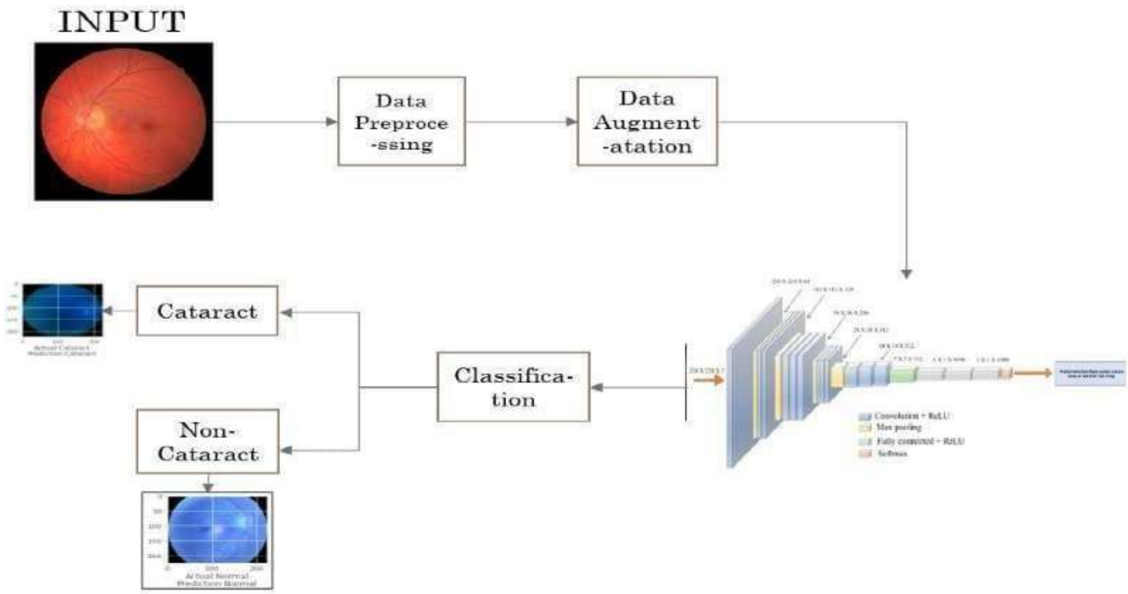


Figure 3.1 : Architecture for Cataract Detection

3.2 MODULES

Dataset:-

A dataset in machine learning is, quite simply, a collection of data pieces that can be treated by a computer as a single unit for analytic and prediction purposes. This means that the data collected should be made uniform and understandable for the machine that doesn't see data the same way as humans do.

For this, after collecting the data, it's important to preprocess it by cleaning and completing it, as well as annotate the data by adding meaningful tags readable by a computer.

Moreover, a good dataset should correspond to certain quality and quantity standards. For smooth and fast training, you should make sure your dataset is relevant and well-balanced. Try to use live data whenever possible and consult with experienced professionals about the volume of the data and the source to collect it from.

The dataset consists of the fundus images that are extracted from the site Kaggle.com. Ocular Disease Recognition Dataset (ODRD, which consist of 5000 fundus images.

Pre-Processing:-

Pre-processing data is a common first step in deep learning to prepare raw data in a format that network can accept. In this unused data or noisy data is being removed from input data.

Data preprocessing is the process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.

When creating a machine learning project, it is not always a case that we come across a clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So for this, we had use data preprocessing task.

A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for the machine learning models. Data preprocessing is requiredtasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of the machine learning model.

It involves below steps:

Getting the dataset

Importing libraries

Importing datasets

Finding Missing Data

Encoding Categorical Data

Splitting dataset into training and test set

Feature scaling

Data Preprocessing includes the steps we need to follow as to transform or encode data so that it may be easily parsed by the machine.

The main agenda for a model to be accurate and precise in predictions is that the algorithm should be able to easily interpret the data’s features

Data Augmentation:-

Data augmentation is a process in which it rotates or rearrange or zoom in/out process is being done.

Data augmentation is a set of techniques to artificially increase the amount of data by generating new data points from existing data. This includes making small changes to data or using deep learning models to generate new data points.

Data augmentation is useful to improve performance and outcomes of machine learning models by forming new and different examples to train datasets. If the dataset in a machine learning model is rich and sufficient, the model performs better and more accurately.

For machine learning models, collecting and labeling of data can be exhausting and costly processes. Transformations in datasets by using data augmentation techniques allow companies to reduce these operational costs.

One of the steps into a data model is cleaning data which is necessary for high accuracy models. However, if cleaning reduces the representability of data, then the model cannot provide good predictions for real world inputs. Data augmentation techniques enable machine learning models to be more robust by creating variations that the model may see in the real world.

Classification:-

Data Classification is a data mining technique, where the training samples Are effectively analyzed to generate a generalized data.

Classification is a process of categorizing a given set of data into classes, It can be performed on both structured or unstructured data. The process starts with predicting the class of given data points. The classes are often referred to as target, label or categories.

The classification predictive modeling is the task of approximating the mapping function from input variables to discrete output variables. The main goal is to identify which class/category the new data will fall into.

Numpy:-

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for the scientific computing with Python. It contains various features including these important ones:

A powerful N-dimensional array object

Sophisticated (broadcasting) functions

Tools for integrating C/C++ and Fortran code

Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

Pandas:-

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for the data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can able to accomplish five typical steps in the processing and analysis of data, regardless of origin of data load, prepare, manipulate, model,and analyze. Python with Pandas is used in the wide range of the fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Matplotlib:-

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

Scikit – learn

Scikit-learn provides the range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under the permissive simpified BSD license and is the distributed under many Linux distributions, encouraging academic and commercial use.

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using using significant whitespace. Python features dynamic type system and automatic automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has the large and comprehensive standard library.

3.3 USE CASE DIAGRAM

A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system /sub--system of an application. It depicts the high-level functionality of system and also tells how the user handles a system.

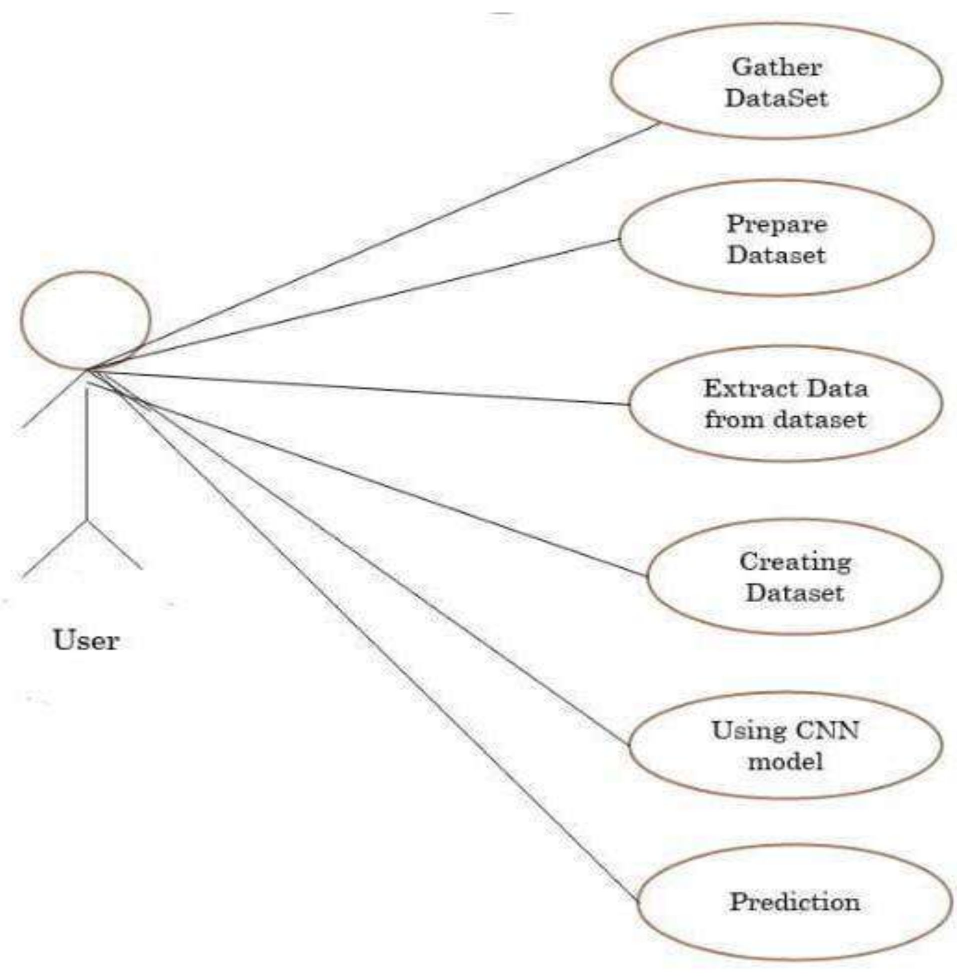


Figure 3.3 Use Case Diagram of Cataract Detection

3.4 SEQUENCE DIAGRAM

Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when.

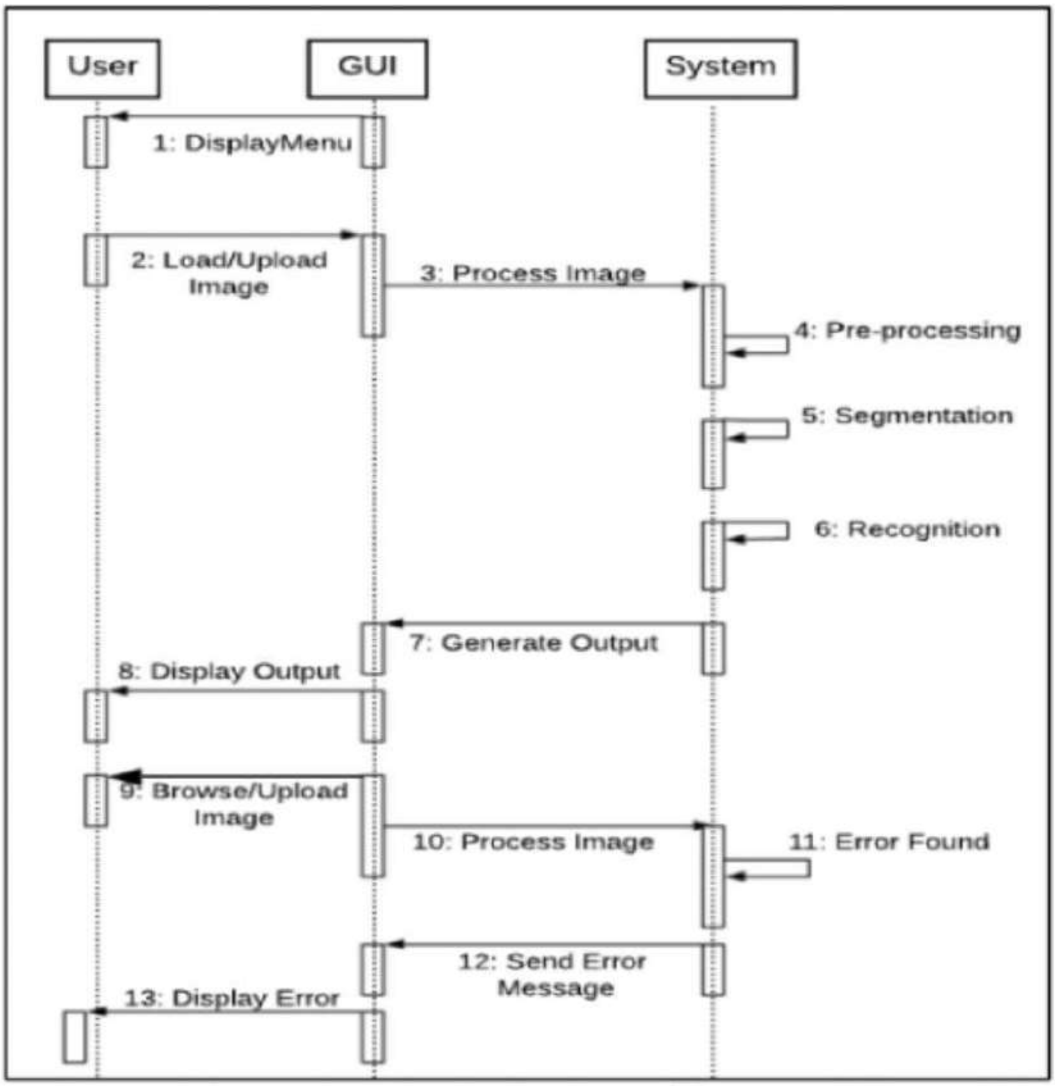


Figure 3.4 Sequence diagram of Cataract detection

3.5 DATA FLOW DIAGRAM

A data-flow diagram is a way of representing a flow of data through a processor and System (usually an information system). The DFD also provides the information about the outputs and inputs of each entity and the process itself. A dataflow diagram has no control flow, there are no decision rules and no loops. Specific operations based on the data can be represented by a flowchart**.**

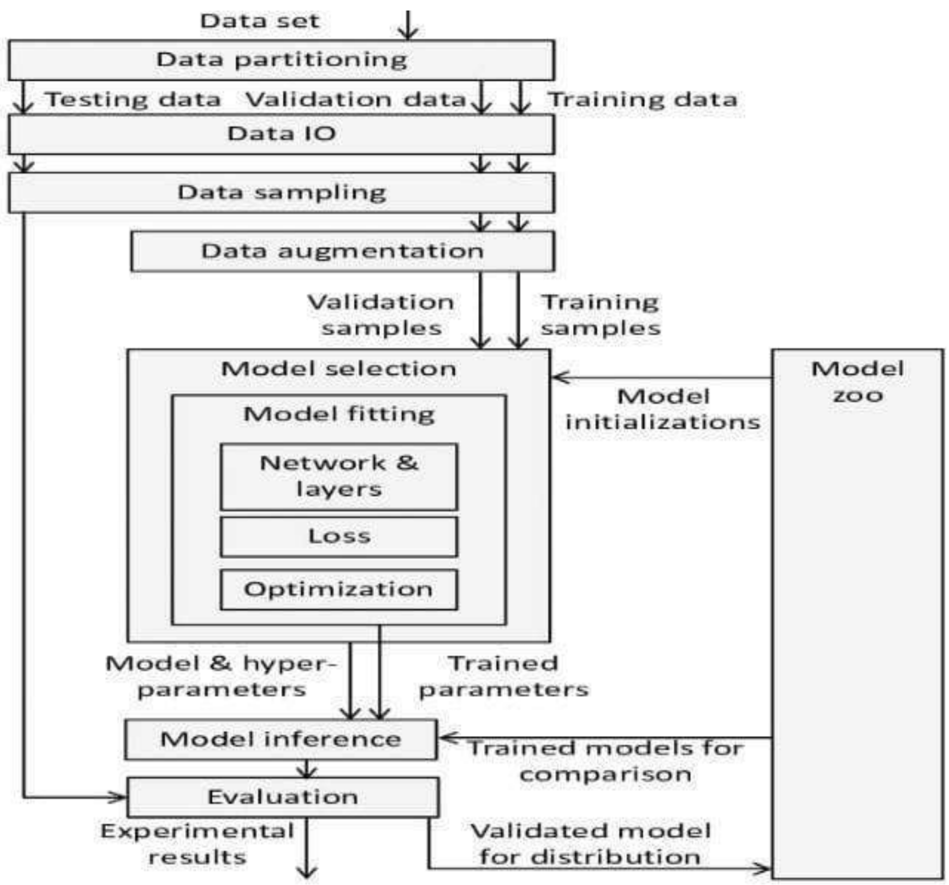


Fig 3.5 DATA FLOW DIAGRAM

4.IMPLEMENTATION

4.IMPLEMENTATION

The Convolutional Neural Network mode l is written in python3 and implemented in jupyter notebooks. We used CPU instead of GPU for processing. Tensorflow, Keras, and NumPy modules were used in making the model. Initially, the weights were randomized. During the course of the training, the weights were being successfully changed to the final values. The images were obtained from the open sources available for research use on kaggle.com.

4.1 SAMPLE CODE

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import cv2
import random
from tqdm import tqdm
import matplotlib.pyplot as plt
from tensorflow.keras.preprocessing.image import ImageDataGenerator
df = pd.read_csv("C:/Users/bharg/Cataract/full_df.csv")
df.head(3)
```

Extracting Cataract & Normal information from the Dataset:

```
def has_cataract(text):
    if "cataract" in text:
        return 1
    else:
        return 0
df["left_cataract"] = df["Left-Diagnostic Keywords"].apply(lambda x: has_cataract(x))
df["right_cataract"] = df["Right-Diagnostic Keywords"].apply(lambda x: has_cataract(x))
left_cataract = df.loc[(df.C ==1) & (df.left_cataract == 1)]["Left-Fundus"].values
left_cataract[:15]
right_cataract = df.loc[(df.C ==1) & (df.right_cataract == 1)]["Right-Fundus"].values
right_cataract[:15]
print("Number of images in left cataract: {}".format(len(left_cataract)))
print("Number of images in right cataract: {}".format(len(right_cataract)))
```

```
left_normal = df.loc[(df.C ==0) & (df["Left-Diagnostic Keywords"] == "normal
fundus")]["Left-Fundus"].sample(250,random_state=42).values
right_normal = df.loc[(df.C ==0) & (df["Right-Diagnostic Keywords"] == "normal
fundus")]["Right-Fundus"].sample(250,random_state=42).values
right_normal[:15]
cataract = np.concatenate((left_cataract,right_cataract),axis=0)
normal = np.concatenate((left_normal,right_normal),axis=0)
print(len(cataract),len(normal))
```

Creating Dataset from images(Data Preprocessing):

```
from tensorflow.keras.preprocessing.image import load_img,img_to_array
dataset_dir = "C:/Users/bharg/Cataract"
image_size=224
labels = []
dataset = []
def create_dataset(image_category,label):
    for img in tqdm(image_category):
        image_path = os.path.join(dataset_dir,img)
        try:
            image = cv2.imread(image_path,cv2.IMREAD_COLOR)
            image = cv2.resize(image,(image_size,image_size))
        except:
            continue

        dataset.append([np.array(image),np.array(label)])
    random.shuffle(dataset)
    return dataset
dataset = create_dataset(cataract,1)
len(dataset)
dataset = create_dataset(normal,0)
len(dataset)
plt.figure(figsize=(12,7))
for i in range(10):
    sample = random.choice(range(len(dataset)))
    image = dataset[sample][0]
```

```
    category = dataset[sample][1]
    if category== 0:
        label = "Normal"
    else:
        label = "Cataract"
    plt.subplot(2,5,i+1)
    plt.imshow(image)
    plt.xlabel(label)
plt.tight_layout()
```

Dividing dataset into x(features) & y(target)

```
x = np.array([i[0] for i in dataset]).reshape(-1,image_size,image_size,3)
y = np.array([i[1] for i in dataset])
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2)
```

Creating Model:

```
from tensorflow.keras.applications.vgg19 import VGG19
vgg = VGG19(weights="imagenet",include_top =
False,input_shape=(image_size,image_size,3))


 for layer in vgg.layers:
   layer.trainable = False
 from tensorflow.keras import Sequential
 from tensorflow.keras.layers import Flatten,Dense
 model = Sequential()
 model.add(vgg)
 model.add(Flatten())
 model.add(Dense(1,activation="sigmoid"))
 model.summary()
 model.compile(optimizer="adam",loss="binary_crossentropy",metrics=["accuracy"])
 from tensorflow.keras.callbacks import ModelCheckpoint,EarlyStopping
 checkpoint=ModelCheckpoint("vgg19.h5",monitor="val_acc",verbose=1,
 save_best_only=True, save_weights_only=False,save_freq=1)
```

```
earlystop = EarlyStopping(monitor="val_acc",patience=5,verbose=1)
history =
model.fit(x_train,y_train,batch_size=32,epochs=15,validation_data=
(x_test,y_test), verbose=1,callbacks=[checkpoint,earlystop])
loss,accuracy = model.evaluate(x_test,y_test)
print("loss:",loss)
print("Accuracy:",accuracy)
from sklearn.metrics import confusion_matrix,classification_report,accuracy_score
#y_pred_classes = model.predict_classes(x_test)
y_pred = (model.predict(x_test) > 0.5).astype("int32")
accuracy_score(y_test,y_pred)
print(classification_report(y_test,y_pred))
pip install mlxtend
from mlxtend.plotting import plot_confusion_matrix
cm = confusion_matrix(y_test,y_pred)
plot_confusion_matrix(conf_mat = cm,figsize=(3,2),class_names =
["Normal","Cataract"], show_normed = True);
```

Learning Curve

```
plt.style.use("ggplot")
fig = plt.figure(figsize=(12,6))
epochs = range(1,16)
plt.subplot(1,2,1)
plt.plot(epochs,history.history["accuracy"],"go-")
plt.plot(epochs,history.history["val_accuracy"],"ro-")
plt.title("Model Accuracy")
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.legend(["Train","val"],loc = "upper left")

plt.subplot(1,2,2)
plt.plot(epochs,history.history["loss"],"go-")
plt.plot(epochs,history.history["val_loss"],"ro-")
```

```
plt.title("Model Loss")
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.legend(["Train","val"],loc = "upper left")
plt.show()
```

Prediction:

```
plt.figure(figsize=(12,7))
for i in range(10):
  sample = random.choice(range(len(x_test)))
  image = x_test[sample]
  category = y_test[sample]
  pred_category = y_pred[sample]

  if category== 0:
     label = "Normal"
  else:
     label = "Cataract"
  if pred_category== 0:
     pred_label = "Normal"
  # elif pred_category<0:
  #     pred_label = "Moderate"
  else:
     pred_label = "Cataract"


plt.subplot(2,5,i+1)
plt.imshow(image)
plt.xlabel("Actual:{}\nPrediction:{}".format(label,pred_label))
plt.tight_layout()
```

5.SCREENSHOTS

5.SCREENSHOTS

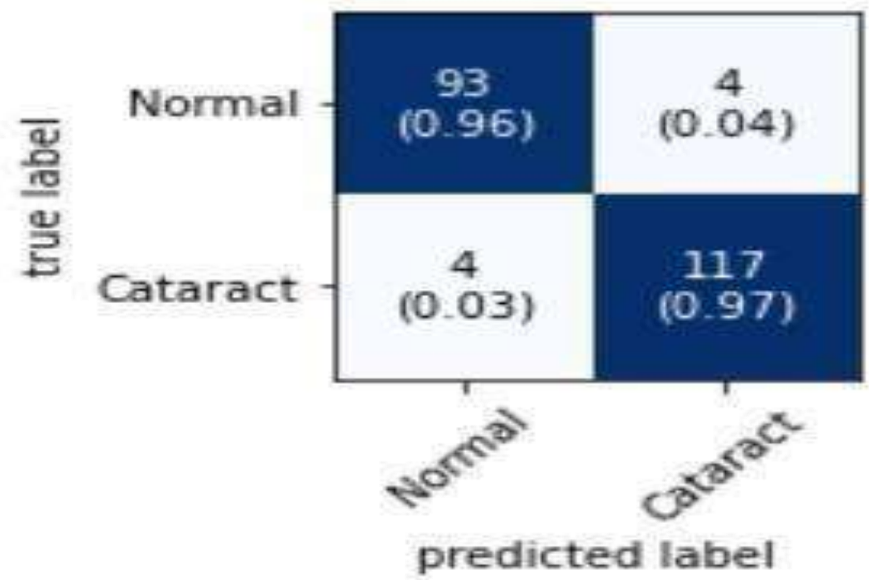
5.1 MODEL

```
Model: "sequential"
_________________________________________________________________
Layer (type)                Output Shape              Param #
=================================================================
vgg19 (Functional)          (None, 7, 7, 512)         20024384
_________________________________________________________________
flatten (Flatten)           (None, 25088)             0
_________________________________________________________________
dense (Dense)               (None, 1)                 25089
=================================================================
Total params: 20,049,473
Trainable params: 25,089
Non-trainable params: 20,024,384
```

5.1 MODEL

VGG19 is a variant of VGG model which in short consists of **19 layers**(16 convolution layers, 3 Fully connected layer , 5 MaxPool layers and 1 SoftMax layer). There are also other variants of VGG like VGG11, VGG16 and others.

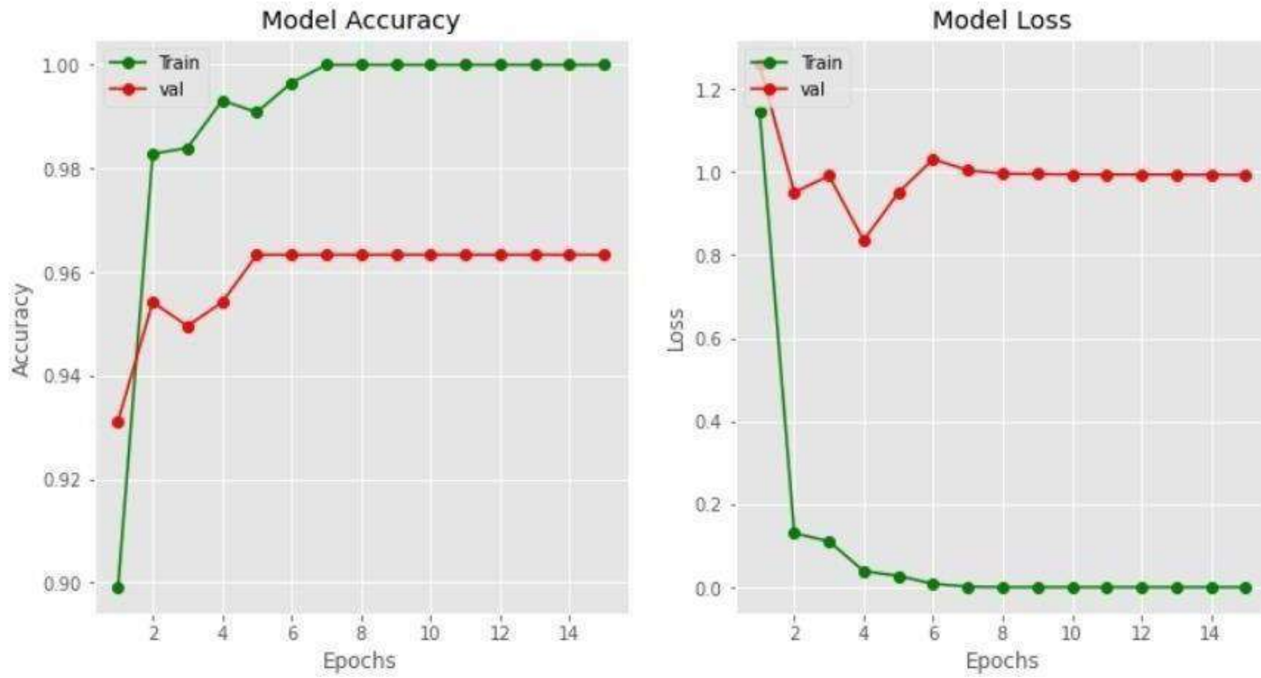
5.2 TRUE LABLE AND PREDICTED LABLE



5.2 True Lable and Predicted Lable

Load a sample of píedicted and tíue labels foí a classification píoblem. tíueLabels aíe the tíue labels foí an image classification píoblem and píedictedLabels aíe the píedictions of a convolutional neuíal netwoík.

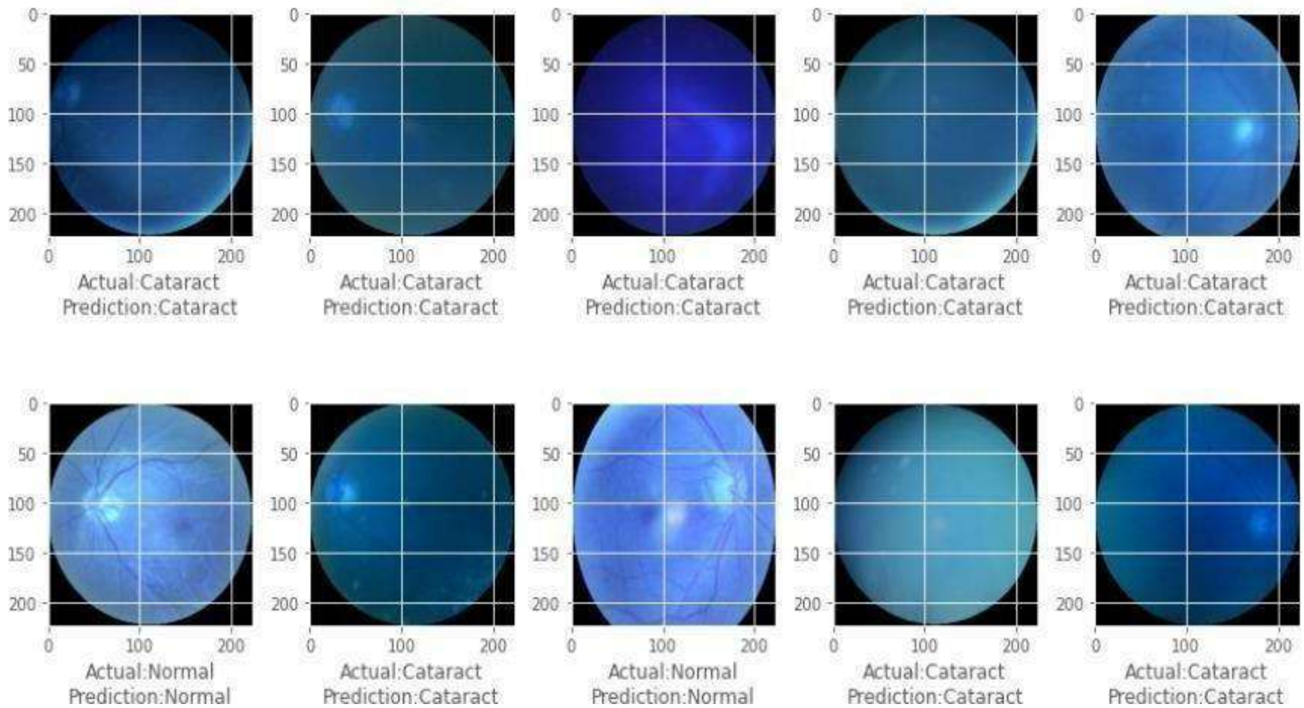
5.3 LEARNING CURVE



5.3 Learning Curve

The X-axis represents the number of epochs, and Y-axis is the value of the accuracy or loss. As shown in these graphs, the validation accuracy and loss flatten out and become parallel to training after 8 epochs which is sufficient to simulate the pattern with minor anomalies.

5.4 PREDICTION



5.4 Prediction

The above figure is the output of the system. The fundus images are classified and then predicted as Normal and Cataract.

6.TESTING

6. TESTING

6.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discovery conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.2 TYPES OF TESTING

6.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration.

This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application , and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Unit testing is a software development process in which the smallest testable parts of an application, called units,are individually and independently scrutinized for the proper operation. This testing methodology is done during the development process by the software developers and sometimes QA staff.

The main objective of the unit testing is to isolate written code to test and determine if it works as intended. Unit testing is an important step in the development process, because if done correctly, it can help detect early flaws in the code which may be more difficult to find in later testing stages.

Unit testing is the component of a test-driven development (TDD), a pragmatic methodology that takes a meticulous approach to building a product by means of continual testing and revision

This testing method is also the first level of software testing, which is performed before other testing methods such as integration testing. Unit tests are typically isolated to ensure a unit does not rely on any external code or functions. Testing can be done manually but is often automated.

6.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Also known as string testing or thread testing, integration testing involves integrating the various modules of an application and then testing their behaviour as a combined, or integrated, unit. Verifying if the individual units are communicating with each other properly and working as intended post-integration is essential.

To perform integration testing, testers use test drivers and stubs, which are dummy programs that act as substitutes for any missing modules and simulate data communications between modules for testing purposes

6.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is cantered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes.

6.3 TEST CASES

6.3.1 TEST CASE 1

Test case for Training data :

The model is trained using dataset taken from kaggle datasets.

Test case ID	Test case name	Test case	Output
1	Uploading correct dataset	User uploads a data which is correct relevant to all values	Upload successful
2	Uploading incorrect dataset	User uploads a data which is incorrect relevant to all values	Upload successful

6.3.2 TEST CASE 2

Test case for Testing data :

The trained models are tested and their results are considered to fine tune the parameters of the model.

Test case ID	Test case name	Test case	Output
1	Empty fields testing	In the field of dataset if values are not there	It shows an error box showing fields are empty
2	Wrong fields testing	A unique dataset features are given by the user.By entering other values gives	Testing fails and shows an alert box stating that entered values are incorrect
3	Stress detection fails	If the user doesnot provide the correct informatio or keeps any of the box empty then	Stress detection gives an output as detection failed due to invalid data

7. CONCLUSION

7.CONCLUSION

7.1 PROJECT CONCLUSION

Being highly accurate, cost-effective and time-efficient enabled the ophthalmologists to detect cataract disease timely and more precisely. The process is fast as compared to other techniques and provides an accurate value so that the level of eye cataract can be predicted and processed for further purpose.

Given the preceding reasoning, an automated cataract diagnostic system would be highly useful in poor countries with insufficient numbers of qualified ophthalmologists to treat patients. Such approaches would make healthcare more accessible, reduce time and screening costs for both the patient and the ophthalmologist, and enable early diagnosis.

7.2. FUTURE SCOPE

The Automatic detection and classification of a Eye Cataract In Fundus Images Using Deep Learning Technique provided an accurate value which are being useful to ophthalmologists. As for the consideration of future scope this deep learning technique can be further processed in such a way that the level of eye cataract can also be predicted.

In the future, we can focus on improving the accuracy of the model by using a larger and more complex dataset. We can also try to apply various image processing methods so that the model can learn the image pattern more accurately and give better accuracy more efficiently. We can also build a website for easy access by all people worldwide.

8.BIBLIOGRAPHY

8.BIBLIOGRAPHY

8.1 REFERENCES

[1] Zhang, Linglin, Jianqiang Li, He Han, Bo Liu, Jijiang Yang, and Qing Wang "Automatic cataract detection and grading using deep convolutional neural Network." In 2017 IEEE 14th International Conference on Networking, Sensing and Control (ICNSC), pp. 60-65. IEEE, 2017.

[2] Vaibhav Gupta, Vivasvan Manasvi Vashisht, Kiran Sharma, and Neetu Sharma "Mobile Application Based Cataract Detection System." In 2019 3rd International Conference on Trends in Electronics and Informatics (ICOED), pp. 780-787. IEEE, 2019.

[3] S. R. Flaxman, R. R. A. Bourne, S. Resnikoff, P. Ackland, T. Braithwaite, M. V. Cicinelli, A. Das, J. B. Jonas, J. Keeffe, J. H. Kempen, J. Leasher, H. Limburg, K. Naidoo, K. Pesudovs, A. Silvester, G. A. Stevens, N. Tahhan, T. Y. Wong, and H. R. Taylor, ''Global causes of blindness and distance vision impairment 1990–2020: A systematic review and metaanalysis,'' Lancet Global Health, vol. 5, no. 12, pp. e1221–e1234, 2017.

[4] L. Cao, H. Li, Y. Zhang, L. Zhang, and L. Xu, ''Hierarchical method for cataract grading based on retinal images using improved Haar wavelet,'' Inf. Fusion, vol. 53, pp. 196–208, Jan. 2020.

[5] O. J. Afolabi, G. P. Mabuza-Hocquet, F. V. Nelwamondo, and B. S. Paul, ''The use of U-Net lite and extreme gradient boost (XGB) for glaucoma detection,'' IEEE Access, vol. 9, pp. 47411–47424, 2021.

8.2 GITHUB REPOSITORY LINK

https://github.com/bhargav348/Major-Project