A

Major Project Report

On

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

(Submitted in partial fulfillment 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 DEP LEARNING" being submitted by R. BHARGAVA CHARY(187R1A05A9), S.PRATHYUSHA(187R1A05B6), CH. NAGASAI TEJA (187R1A0568), G.SHIRISHA(18C21A0553)in partial fulfillment 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. G. Madhukar
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) Mr. J. Narasimha Rao, Dr. M. Varaprasad 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 clouding of the lens in the eye which is painless and developed gradually over a long period. Cataract is an eye disorder which occurs when some of protein at lens clumped together that makes it dull and increases opacity of the lens, causing some loss of vision. Most of the cataract is related to aging. In this work we presented VGG-19 Algorithm, a deep neural network for automatically detecting cataracts in fundus 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 classified the results where we have achieved Accuracy of 96% and loss of 0.993.

LIST OF FIGURES/TABLES

LIST OF SCREENSHOTS

TABLE OF CONTENT

1. INTRODUCTION

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.

1.2 PROJECT PURPOSE

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

The stack of the model can be an android app or any built device such as CATRA [6], and here we concentrate on the method of implementation.

2.SYSTEM ANALYSIS

2. 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 92.1% for cataract detection and finally the system Is trained to classify an input eye image as cataract or normal eye.

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.

ResNet 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

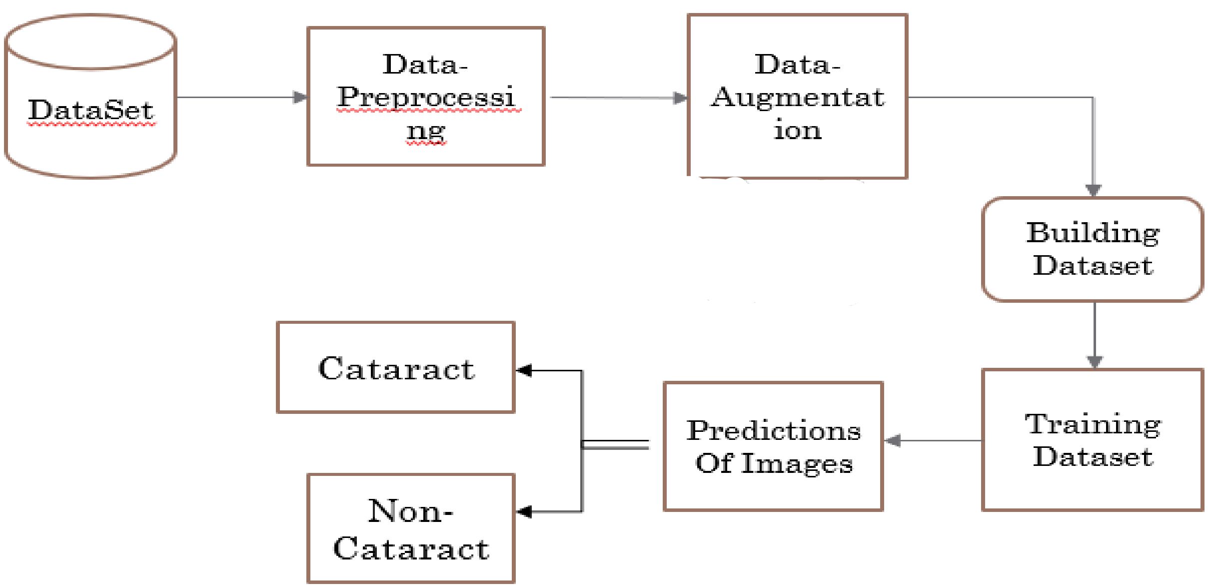
- The techniques used in existing system are not able to produce accurate values.
- It is complex to identify the cataract.
- It is an expensive procedure.

2.3 PROPOSED SYSTEM

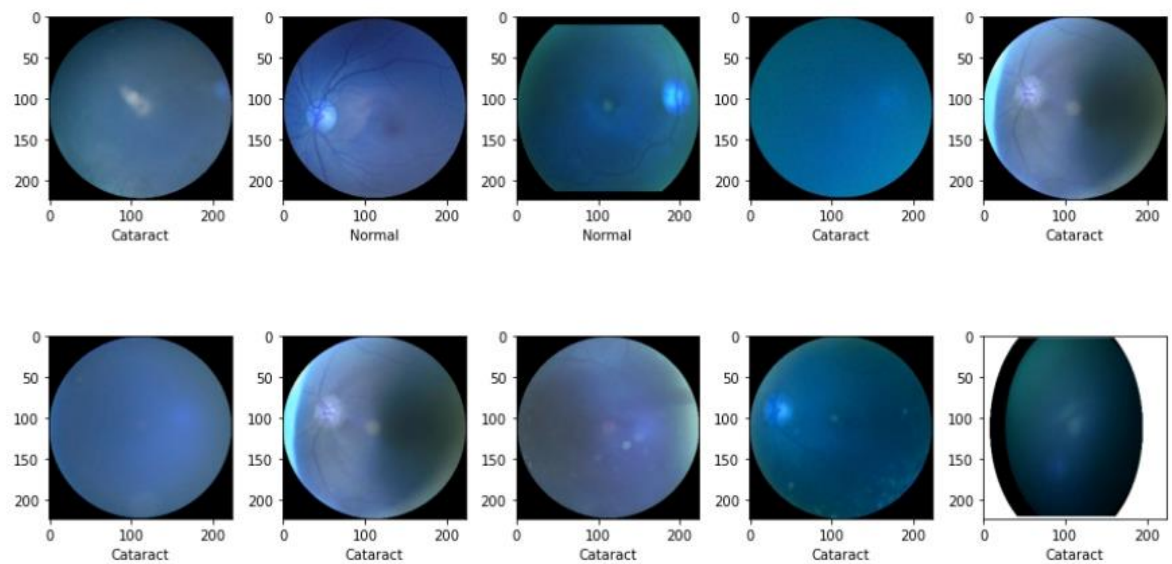
We presented an automated cataract detection system namely VGG-19 based on deep learning and the dataset is rearranged, preprocessed and augmented to improve the dataset to feed deep network.

It classifies the patients into two groups:

Cataract or Non-Cataract conditions.



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



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

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

- RAM : 8GB and Higher
- Processor : Intel i5
- Hard Disk : 500GB Minimum

2.5.2 SOFTWARE REQUIREMENTS

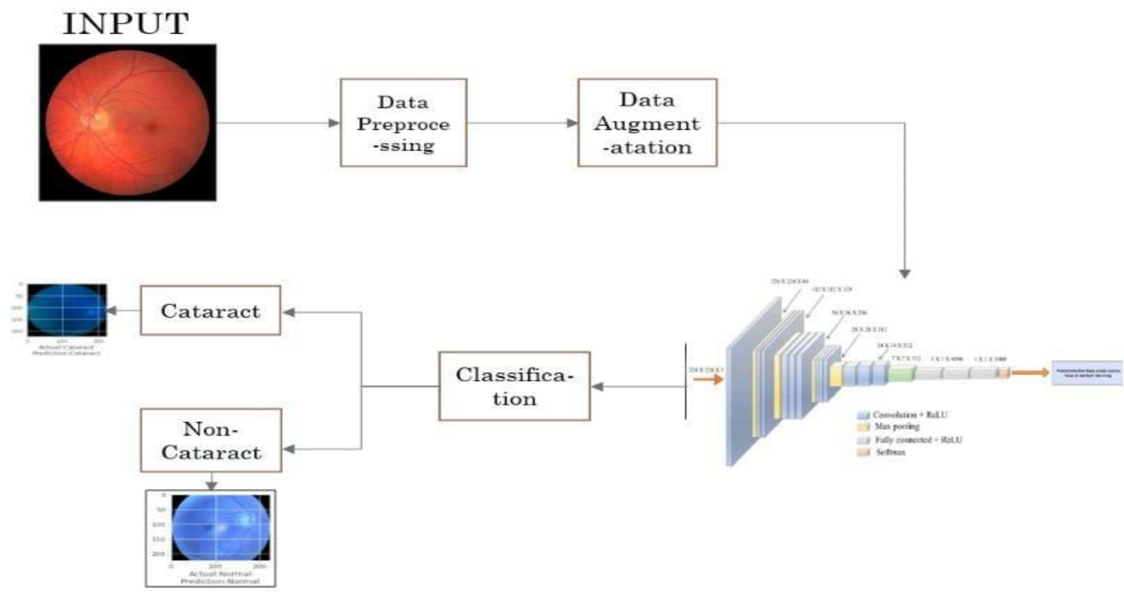
Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements.

- Operating System : Windows 10
- Python IDE : Python 3, Anaconda

3.ARCHITECTURE

3.ARCHITECTURE

3.1 PROJECT ARCHITECTURE



3.2 MODULES

Dataset:-

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

Pre-Processing:-

Preprocessing 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.

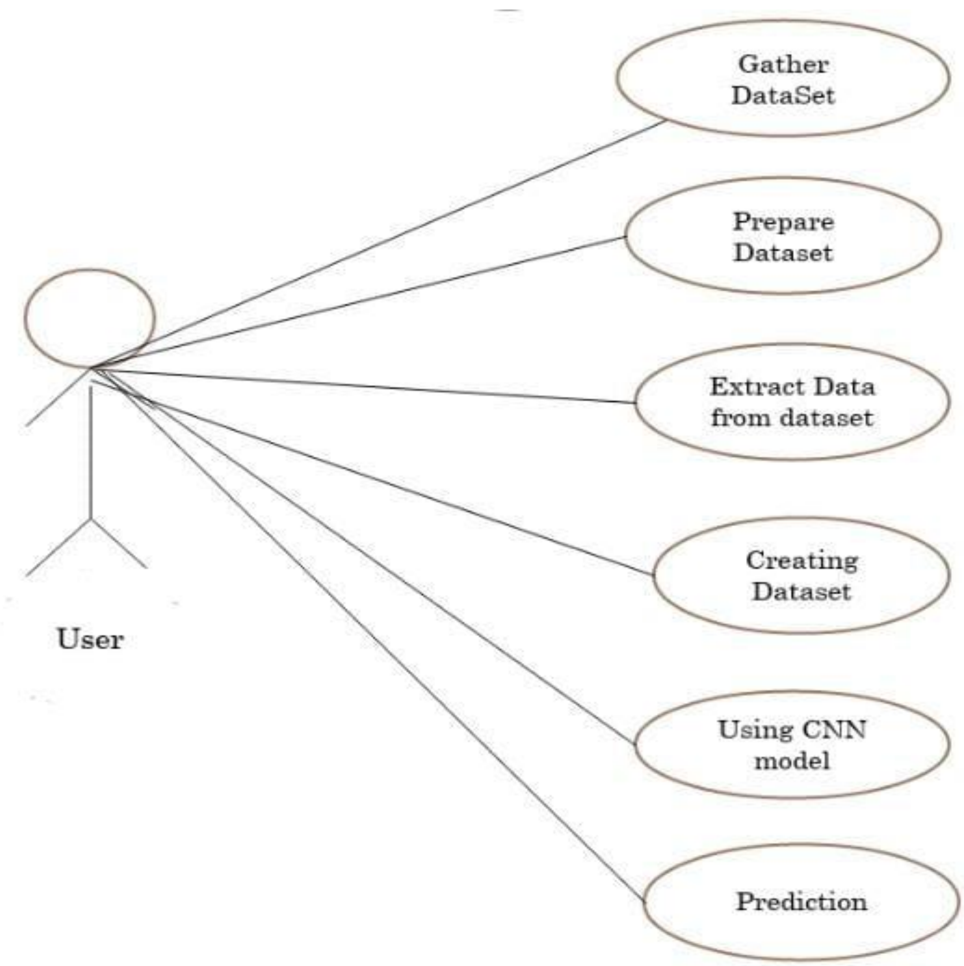
Data Augmentation:-

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

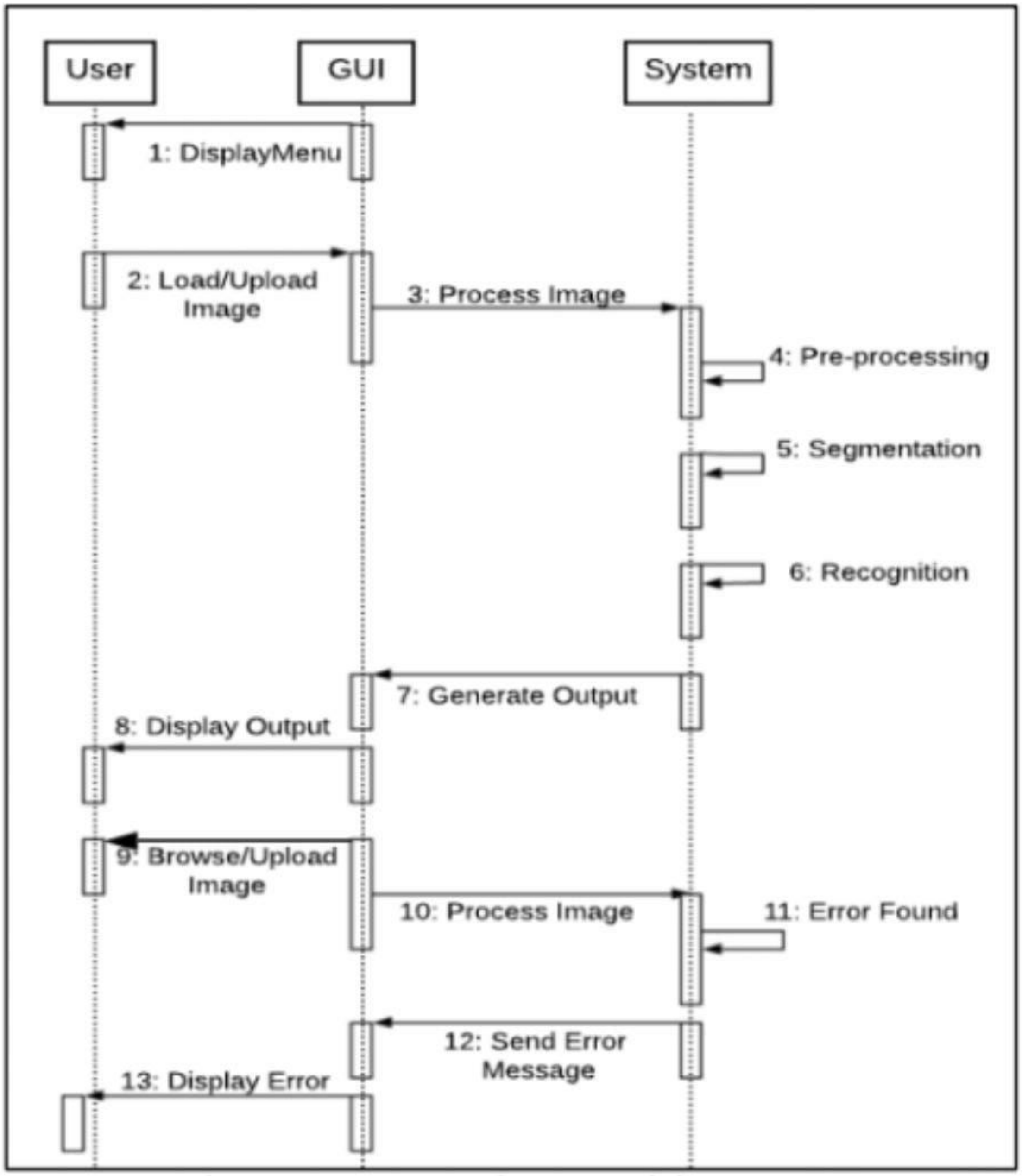
Classification:-

Classifying the fundus images into Cataract or Non cataract and also classifying that the cataract is at which level.

3.3.USE CASE DIAGRAM



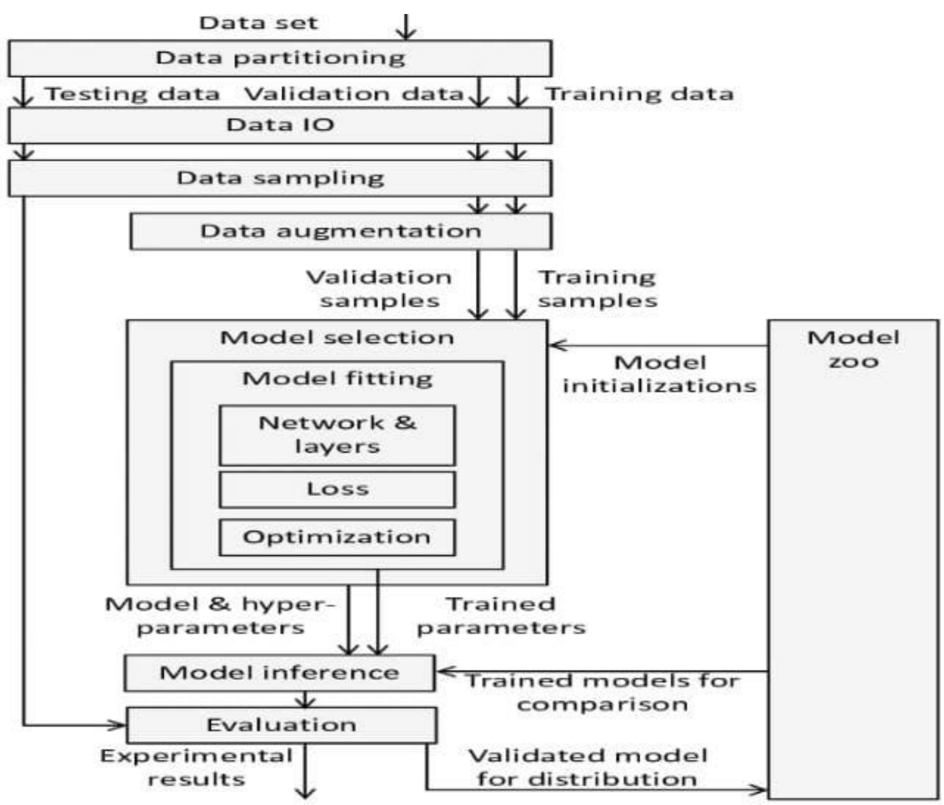
3.4. SEQUENCE DIAGRAM



3.4 SEQUENCE DIAGRAM

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 data-flow 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.



3.5 DATA FLOW DIAGRAM

4.IMPLEMENTATION

4.IMPLEMENTATION

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
import os
for dirname, _, filenames in os.walk('/Cataract'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
df = pd.read_csv("C:/Users/bharg/C-
Project/Evaluation_Set/Evaluation_Set/full_df.csv")
df.head(3)

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))

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()

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)
```

```
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=["ac
curacy"])

from tensorflow.keras.callbacks import ModelCheckpoint,EarlyStopping
checkpoint =
ModelCheckpoint("vgg19.h5",monitor="val_acc",verbose=1,save_best_onl
y=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=5,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))

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);

plt.style.use("ggplot")
fig = plt.figure(figsize=(12,6))
epochs = range(1,6)
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()

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

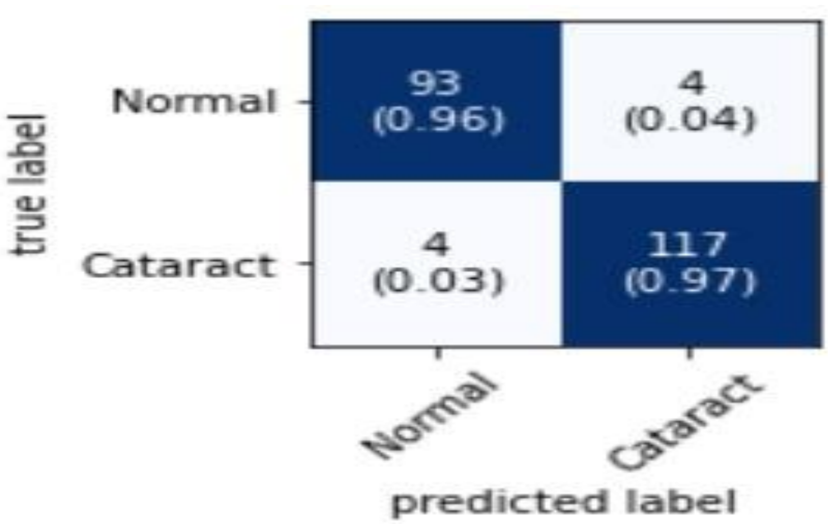
5.SCREENSHOTS

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
```

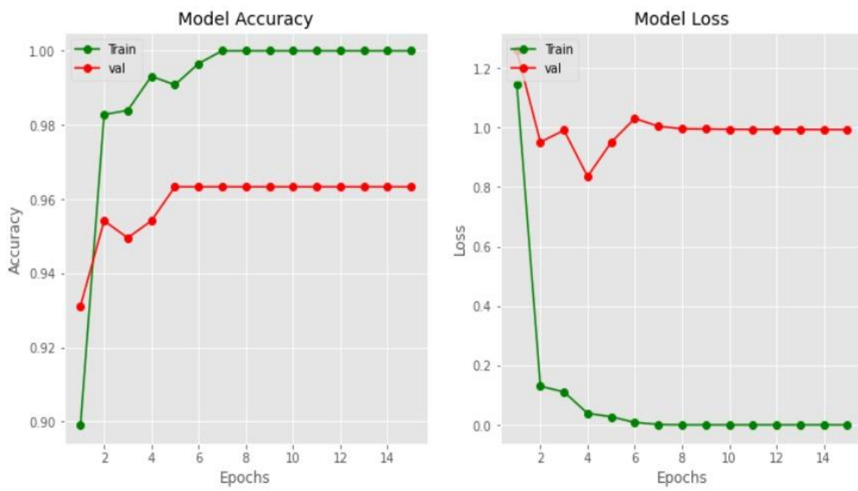
5.1 MODEL

5.2 TRUE LABLE AND PREDICTED LABLE



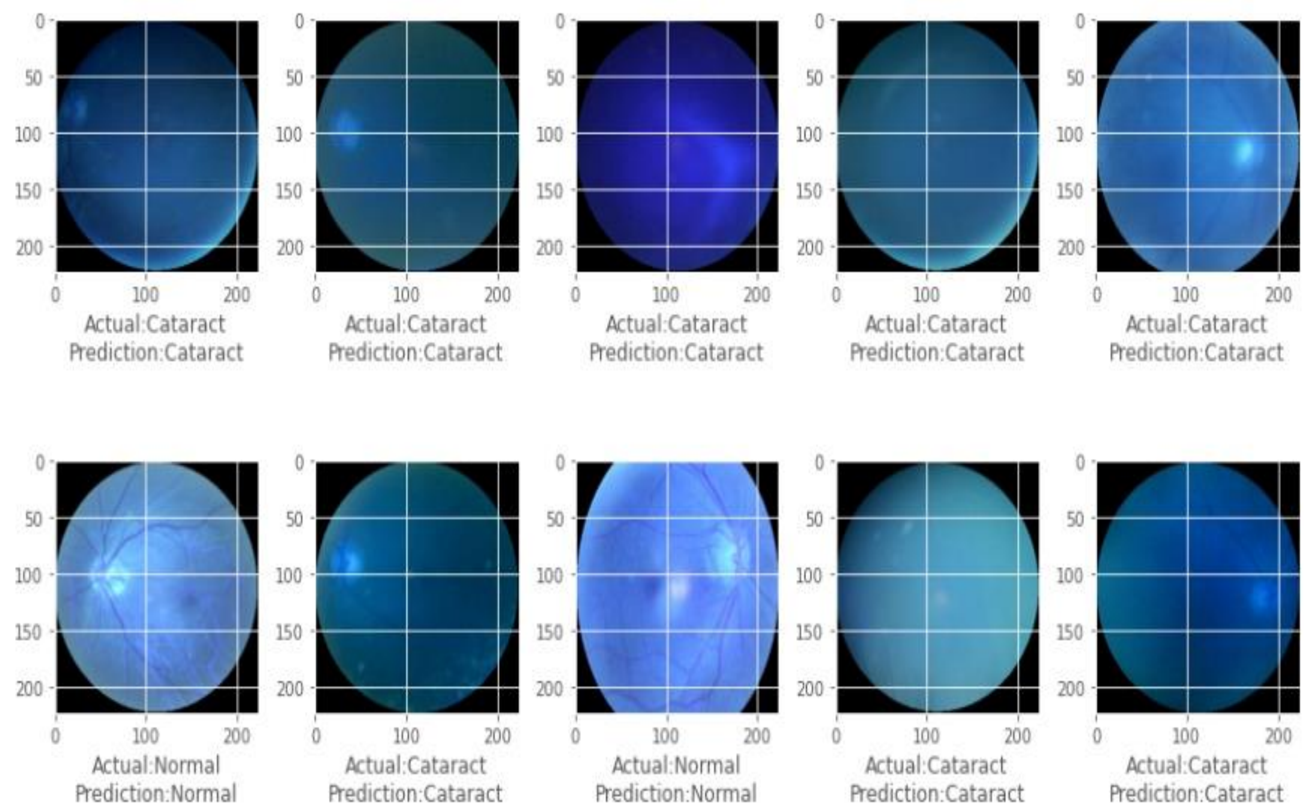
5.2. True Lable and Predicted Lable

5.2 LEARNING CURVE



5.3. Learning Curve

5.4 PREDICTION



5.4. Prediction

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.

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.

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.

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.

7.2. FUTURE SCOPE

8. BIBILOGRAPHY

8. BIBILOGRAPHY

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.