** MOBILE APPLICATION FOR CATARACT DETECTION**

*EPICS PROJECT REPORT submitted in partial fulfillment of the requirements for the Award of the Degree of*

## BACHELOR OF TECHNOLOGY

**In**

**INFORMATION TECHNOLOGY**

**By**

**Aluri Charan (208W1A1268)**

**Kolluri Mounav (208W1A1291)**

**Panchumarthi Sai Srinivas (208W1A12A4)**

*Under the Guidance of*

**Dr.T . Anuradha, Ph.D**

**Professor**



**DEPARTMENT OF INFORMATION TECHNOLOGY V R SIDDHARTHA ENGINEERING COLLEGE**

**(AUTONOMOUS - AFFILIATED TO JNTU-K, KAKINADA)**

**Approved by AICTE &Accreted by NBA KANURU, VIJAYAWADA-520007 ACADEMIC YEAR**

**(2022-23)**

V.R.SIDDHARTHA ENGINEERING COLLEGE

(Affiliated to JNTUK: Kakinada, Approved by AICTE, Autonomous) (An ISO certified and NBA accredited institution)

Kanuru, Vijayawada – 520007



# CERTIFICATE

This is to certify that this project report titled “MOBILE APPLICATION FOR CATRACT DETECTION” is a bonafide record of work done by **Aluri charan (208W1A1268)** , **Kolluri Mounav (208W1A1291)**, **Panchumarthi Sai Srinivas (208W1A12A4)** under my guidance and supervision is submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Information Technology, **V.R. Siddhartha Engineering College**(Autonomous under JNTUK) during the year **2022-2023.**

### (Dr.T.Anuradha) (Dr. M. Suneetha)

Professor Professor & Head

Dept. of Information Technology Dept. of Information Technology

### EXTERNAL EXAMINER SIGNATURE

*Date of examination:*

## 

## ACKNOWLEDGEMENT

## First and foremost, I sincerely salute our esteemed institution VR SIDDHARTHA ENGINEERING COLLEGE for giving me this opportunity to fulfilling my project. I am grateful to our principal Dr. A.V. RATNA PRASAD, for his encouragement and support all through the way of my project.

## On the submission of this project report, I would like to extend my honor to Dr .M. Suneetha, Head of the Department, IT for her constant motivation and support throughout my work.

## I feel glad to express my deep sense of gratefulness to my project guide Dr.T.Anuradha, Professor for her guidance and assistance in completing this project successfully.

## I would also like to convey my sincere indebtedness to all the faculty members, including supporting staff of the department, friends, and family members who bestowed their great effort and guidance at appropriate times without which it would have been very difficult on my part to finish the project work.

# DEPARTMENT OF INFORMATION TECHNOLOGY VELAGAPUDI RAMAKRISHNA SIDDHARTHA ENGINEERING COLLEGE

**PROJECT SUMMARY**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **ITEM** | **DESCRIPTION** |
| 1 | Project Title | Mobile Application For Cataract Detection |
| 2 | Student Names And  Numbers | Aluri charan (208W1A1268)  Kolluri Mounav(208W1A1291)  Panchumarthi Sai Srinvas(208W1A12A4) |
| 3 | Name of The Guide | Dr.T.Anuradha |
| 4 | Name of The Mentor | Dr. Shaik Fatimabi |
| 5 | Research Group | BigData,Data Analytics |
| 6 | Application Area | Health Care |
| 7 | Aim of the Project | The project aims to find out whether a person is suffering from cataract or not |
| 8 | Project Outcomes | Mobile application and a deep learning model That can predict whether a person is suffering from cataract or not |

### Student Signatures

### Signature of the Guide

Dr.T.Anuradha

**TABLE OF CONTENTS**

[ACKNOWLEDGEMENT iii](#_bookmark0)

[LIST OF FIGURES vi](#_bookmark1)

[LIST OF TABLES vii](#_bookmark2)

[ABSTRACT viii](#_bookmark3)

[CHAPTER-1 Introduction 1](#_bookmark4)

1.1 [Origin of the Problem 1](#_bookmark5)

* 1. [Basic definitions and Background 1](#_bookmark6)
  2. Problem Statement with Objectives and Outcomes 2

1.4 Real time Applications of Proposed work 3

[CHAPTER-2 Review of Literature 4](#_bookmark7)

* 1. [Description of Existing Systems 4](#_bookmark8)

2.2 Summary of Literature Study 5

[CHAPTER-3 Proposed Method 6](#_bookmark9)

* 1. Design Methodology 6

3.2 [System Architecture Diagram 8](#_bookmark10)

3.3 Description of Algorithms 9

3.4 [Description of Requirements and Tools 11](#_bookmark11)

CHAPTER-4 Results and Observations 13

* 1. [Stepwise description of Results 13](#_bookmark12)

4.2 result Analysis 15

4.3 [Observations from the work 16](#_bookmark13)

CHAPTER-5 Conclusion and Future work 17

5.1 [Conclusion 17](#_bookmark14)

5.2 [Future study 17](#_bookmark15)

[References 18](#_bookmark16)

[Appendix 19](#_bookmark17)

## LIST OF FIGURES

## Figure Page No

**Figure 3.1:** Architecture Diagram 8

**Figure 3.1.1:** Convolution Neural Network Diagram 9

**Figure 3.2:** Anaconda Prompt 10

**Figure 3.3:** Jupyter Notebook 11

**Figure 3.4:** Android studio 12

**Figure 4.1:** Test the eye 13

**Figure 4.2:** Scan the eye 13

## LIST OF TABLES

**Table Page No**

**Table 2.1:** Literature Review 4

## ABSTRACT

Cataracts are among the most common conditions that people experience as they get older. The most typical signs of cataracts are black spots in front of the eyes, blurred and distorted images, and hazy vision. The rate at which cataracts form is not very quick. It proceeds slowly. Either one or both eyes may be impacted.

Inadequate facilities and a lack of access are the two main issues that are negatively influencing eye care in India. Only 15,000 ophthalmologists are thought to be practising in India. It is now difficult to diagnose a sickness.

The majority of those who have this disease must visit a doctor to determine whether it is present.

In this paper, a technique is suggested and put into practice, and an Android application for smartphones is

produced following the suggested methods that can be used to identify whether a person's eyes have cataracts. The methodology is developed using deep learning and image processing techniques.

**Keywords:** Cataract , Deep Learning, Smartphone , Image Processing

## CHAPTER – 1 INTRODUCTION

In India, there are more than 10 million persons who are blind on both sides. Cataract is the main factor in over 70% of cases. The clear lenses of the human eye become clouded as a result of a disorder called a cataract. Most cataracts gradually worsen over time, frequently denying the patient any hope of saving their eyesight. However, if a cataract is found, it may be surgically removed while preserving vision. As a result, many cases of cataracts go undiagnosed and result in blindness.

Thanks to the increased evaluation of smartphones with detecting capabilities, we now can design and create a range of apps. Particularly, advances in portable technology are creating new opportunities for the delivery of human services. Cellphones and other portable devices, for instance, have significantly changed the modern world.

### 1.1 Origin of the Problem:

### Over the last few decades, Cataract is the main disease that is causing bilateral blindness for many people across the globe. With Indians getting it 10 to 15 years earlier than the western people makes it is important to detect cataracts at an early stage. But the medical facilities available in India are fewer when compared to foreign countries. So developing a cost effective and mostly accessible.

### solution is an urgent requirement

### 1.2 Basic definitions and Background:

**1.2.1 CONVOLUTIONAL NEURAL NETWORKS:** The convolution neural network is a deep

Learning algorithm which is commonly used for image classification and image processing applications this algorithm takes images as input and will assign weights and biases to various features of images and be able to comprehend one from each other. The preprocessing required in this algorithm is less when compared with any supervised algorithms

**1.2.2 ACTIVATION FUNCTION:** The activation function of a node in an artificial neural network determines that node's output given an input or group of inputs. An ordinary integrated circuit may be thought of as a digital network of activation functions that, depending on input, are either "ON" (1) or "OFF" (0). This is comparable to a neural network's linear perceptron. However, only nonlinear activation functions—also referred to as nonlinearities—allow such networks to calculate nontrivial problems using just a few nodes.

**1.2.3 OPTIMIZER:** Optimizers are programmes or techniques that modify the neural network's properties, such as its weights and learning rate, in order to minimise losses. By reducing the function, optimizers can address optimization issues.

**1.2.4 SOFTWARE DEVELOPMENT KIT:** An SDK is a collection of software tools and apps that hardware and software providers offer to developers so they may create applications for certain platforms. SDKs make it simple for developers to connect their applications to a vendor's services.

**1.3 Problem Statement:**

Detection of Cataract through mobile using Deep learning Algorithms

**1.4 SOCIETIAL APPLICATIONS OF PROPOSED WORK**

In India, healthcare is getting increasingly expensive and out of reach for most people. There are extremely few ophthalmologists available in rural locations when it comes to healthcare. As a result, this application may be employed as cataract detection equipment. Through the use of mobile devices, people may examine their eyes, which is making eye screening more accessible to people in rural areas .

## CHAPTER –2 REVIEW OF LITERATURE

This chapter describes the review of literature that we have taken from various papers and considered all the points mentioned in the papers.

### Description of Existing Systems:

The literature review of numerous publications is shown in the table below, along with explanations of the observations we made from each paper.

**Table 2.1:** Literature Review

| **Sno** | **Paper Title** | **Methodology** | **Year** | **Algorithms** | **Result** |
| --- | --- | --- | --- | --- | --- |
| 1 | Detecting Cataract Using Smartphones | Has developed a classifier model based on Support Vector Machine (SVM). They have used luminance transformation and pixel brightness Algorithm to extract lens images features  . | 2021 | Support Vector Machines (SVM) | A mobile app that can detect  Cataract |
| 2 | CataractNet: An Automated Cataract Detection  System Using Deep Learning for Fundus Images | Has developed a Deep neural system called cataractNet, Which is pre trained Convolution neural network model, They have taken nearly 1130 images from the ACHIKO-1 fundus image data set they got an accuracy of nearly 97%. | 2021 | Convolutional Neural Network (CNN) | A Deep Neural Network System called  CataractNet |

.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 3 | Unified Diagnosis Framework for Automated  Nuclear Cataract Grading Based on  Smartphone Slit-Lamp Images | Has developed a classifier model using the Support vector machine (SVM) They trained the model using 819 anterior ocular images captured using a Smartphone based slit light They collected the images from Marked slit lamp picture project (MSLPP) created by the HE Eye specialist Hospital(HESH) China | 2020 | Support Vector Machines | A model that can predict the presence of cataract from slit lamp images |

## 

* 1. **Summary of Literature Study:**

1. Developed an algorithm for luminance transformation from pixels brightness. Using this algorithms they extracted luminance features of eye images that were takes from a smart phone.

They applied watershed transformation for images to pre process them. They adopted support vector

Machine (SVM) as the classification method by this they got an accuracy of 96%.By adopting to this

Luminance based approach results from various smartphones will be same

1. Developed an deep neural network CataractNet, for detecting cataract from fundus images.They have used the loss and activation functions which are tuned to train the network with small kernels, fewer training parameters, and fewer layers. A total of 1130 images were used for training og neural network they have achieved an accuracy of 98.13%
2. Developed an SVM and ShuffleNet based deep learning network for catract detection from slit lamp images.They used the Svm model to detect the cataract severity y evaluating the gray conjugate features of necular region.They have trained the classification algorithm by using 819 anterior ocular images taken from smart phone based slit lamp and they got an accuracy of 93.5%

## CHAPTER-3 PROPOSED METHOD

This chapter describes the architecture diagram and algorithms that we have applied for our project to find the change that happened between two different timestamp images.

**3.1 Design Methodologies:**

In this project we have used python tensorflow library for developing the neural network and used android studio for creating the UI of the Mobile application.The UI is used to take the input from the user and hide the backend processing.

The overall methodology is divided into following steps:

**Step-1:** Collection of images

**Step-2**: Preprocessing of Data

**Step-3:** Classifiaction by deep learing models

**Step-4:** Validation of trained model

**Step 1 – Collection of Images**

The data is collected from google search image and some of the images were collected from github

Some real time data is being collected from nearby hospitals

**Step 2 – Preprocessing of data**

The collected images are cropped and rotated to get the pupil region of eye and each image is cropped into 224\*224 pixels and a total of 8068 images were prepared for training

**Step-3: Classification by deep learning models**

**Creating the convolution base:**

As input CNN takes an tensor of shape (image\_height, image\_width,color\_channele) we are giving the input size as 64\*64\*3 as the we are giving rgb images we have to specify the shape in input\_shape parameter

**Code:**

classifier = Sequential()

#Convolution

classifier.add(Convolution2D(32,3,3, input\_shape=(64,64,3),activation='relu'))

#Pooling

classifier.add(MaxPooling2D(pool\_size=(2,2)))

#Addition of Second convolutional layers and pooling to get more accuracy on the test\_set

classifier.add(Convolution2D(32,3,3,activation='relu'))

classifier.add(MaxPooling2D(pool\_size=(2,2)))

**Adding Dense Layers On Top:**

To complete the model, you will feed the last output tensor from the convolutional base (of shape (4, 4, 64)) into one or more Dense layers to perform classification. Dense layers take vectors as input (which are 1D), while the current output is a 3D tensor. First, you will flatten (or unroll) the 3D output to 1D, then add one or more Dense layers on top.

**Code:**

classifier.add(Flatten())

#Full connection

classifier.add(Dense(128,activation='relu'))

classifier.add(Dense(1,activation='sigmoid'))

**Compiling and Training the model:**

Then the model will be compiled and trainde

**Code:**

model.compile(optimizer='adam',  
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from\_logits=True),  
              metrics=['accuracy'])  
  
history = model.fit(train\_images, train\_labels, epochs=10,   
                    validation\_data=(test\_images, test\_labels))

**Saving the best model:**

The model with the highest accuracy is saved

**Code:**

from google.colab import files

uploaded = files.upload()

from tensorflow.contrib import lite

from google.colab import files

converter = lite.TFLiteConverter.from\_keras\_model\_file( 'C\_and\_N.h5' )

model = converter.convert()

file = open( 'model.tflite' , 'wb' )

file.write( model )

files.download('model.tflite')

**Step-4: Validation of trained model**

Then the trained model is validates with the testing data and the model will be evaluated based on

Different measures like accuracy,validation loss , F-score etc.

### 3.2 System Architecture Diagram:

A picture containing timeline

Description automatically generated

**Figure 3.1:** Architecture Diagram

**3.3 Description of Algorithms:**

-> **Convolution Neural Networks:**

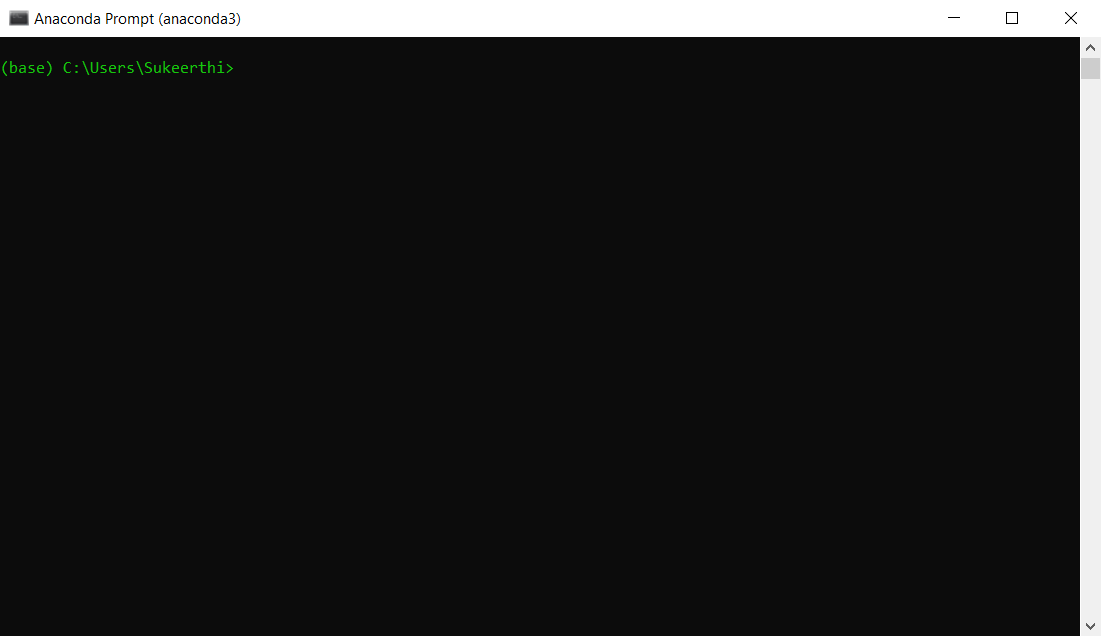
Face detection is one of the most important technologies for driving drowsiness detection based on computer vision. In practical application, driving drowsy detection system not only requires high accuracy, but also fast speed. As we know, deep learning methods, especially the convolutional neural network model, greatly improve the accuracy of image recognition. However, the complex network structure reduces the algorithm speed. In, a Multi-Task Cascaded Convolutional Networks called MTCNN have been designed for face detection. The architecture consists of three sub-networks, instead of a complex network. Each sub-network has less numbers of filters but more discrimination of them, which effectively improves the speed of the algorithm. However, we noticed its performance might be limited by the useless information about the five facial landmarks. Because we just pay attention to the information of the facial in video frames. Thus, we design a new convolutional neural network to detect the face of a driver. It is more efficient thanks to the lightweight architecture and removing five facial landmarks. The architecture of Deep Cascaded Convolutional Neural Network (DCCNN) is shown in Figure 2. In practical application, the proportion of the face in frames is uncertain. Thus, it is an effective approach to resize the original image into different scales. Similar to MTCNN, we build an image pyramid by resizing the original image to different scales, which is the input for three cascaded networks. Network-1 is a fully convolutional network, to obtain a large number of candidate windows and their bounding box regression vectors. Then we use the estimated bounding box regression vectors to calibrate the candidates. Finally, non-maximum suppression(NMS) is employed to merge highly overlapped candidates. After Network-1, All candidates are fed to the next CNN, called network-2, which further reject a larger number of false candidates, performs calibration with bounding box regression, and NMS candidate merge. Network-3 is a CNN similar to network-2, but it has more convolutional layers and pooling layers so that it can describe the face in detail. The output of network-3 is the facial bounding box with high confidence. The relationship of the three sub-networks is cascaded. In DCCNN, the output of the previous sub-network acts as the input to the next sub-network. By building the image pyramid, the face of a driver can be detected accurate no matter how larger proportion of the face in frames.



**Figure 3.2:** Convolution Neural Network Diagram

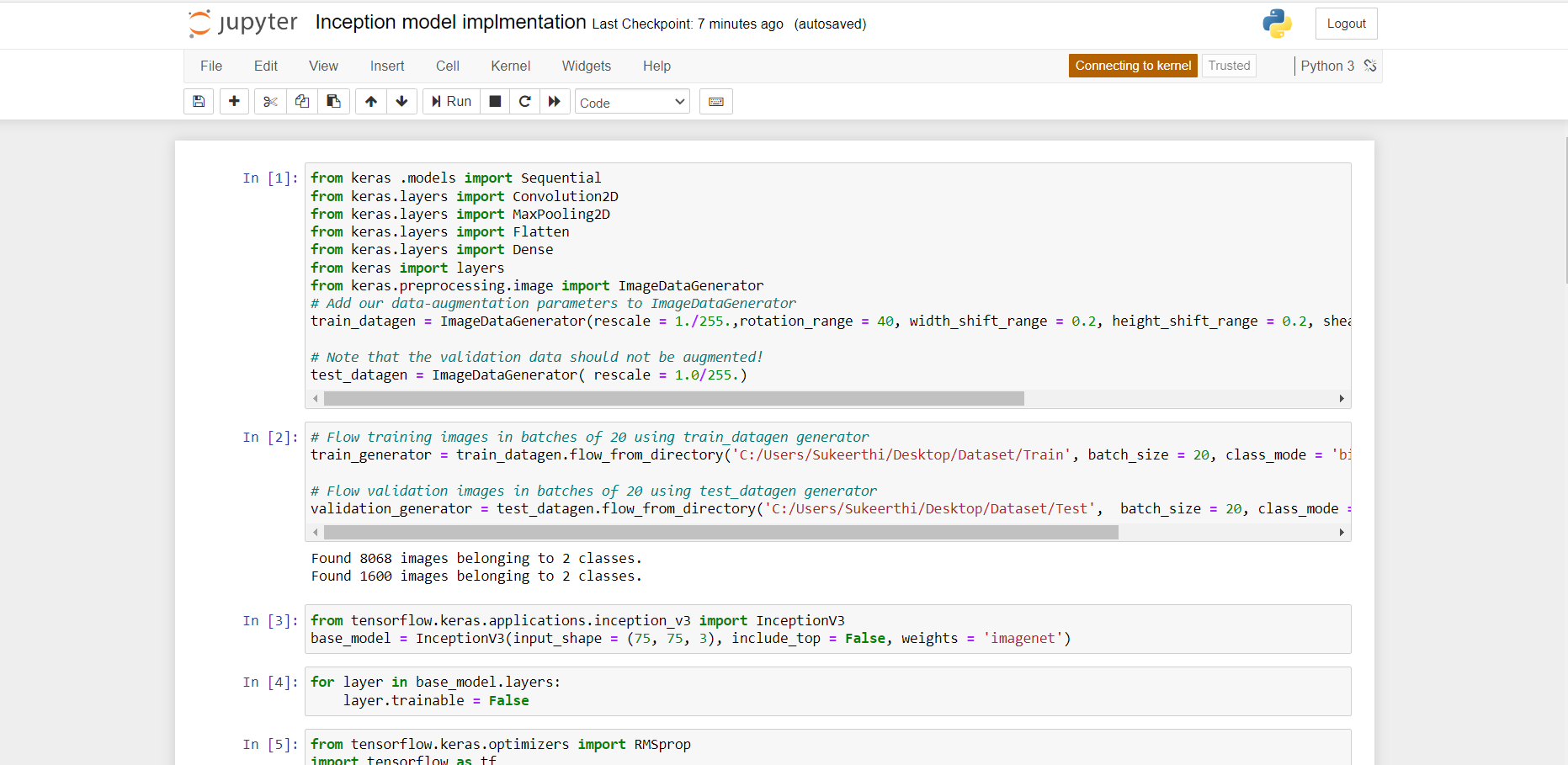
* 1. **Description of Requirements and Tools:**

**Anaconda Distribution:** Anaconda is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS. It is developed and maintained by Anaconda .Package versions in Anaconda are managed by the package management system conda.[9] This package manager was spun out as a separate open-source package as it ended up being useful on its own and for things other than Python.[10] There is also a small, bootstrap version of Anaconda called Mini conda, which includes only conda, Python, the packages they depend on, and a small number of other packages.



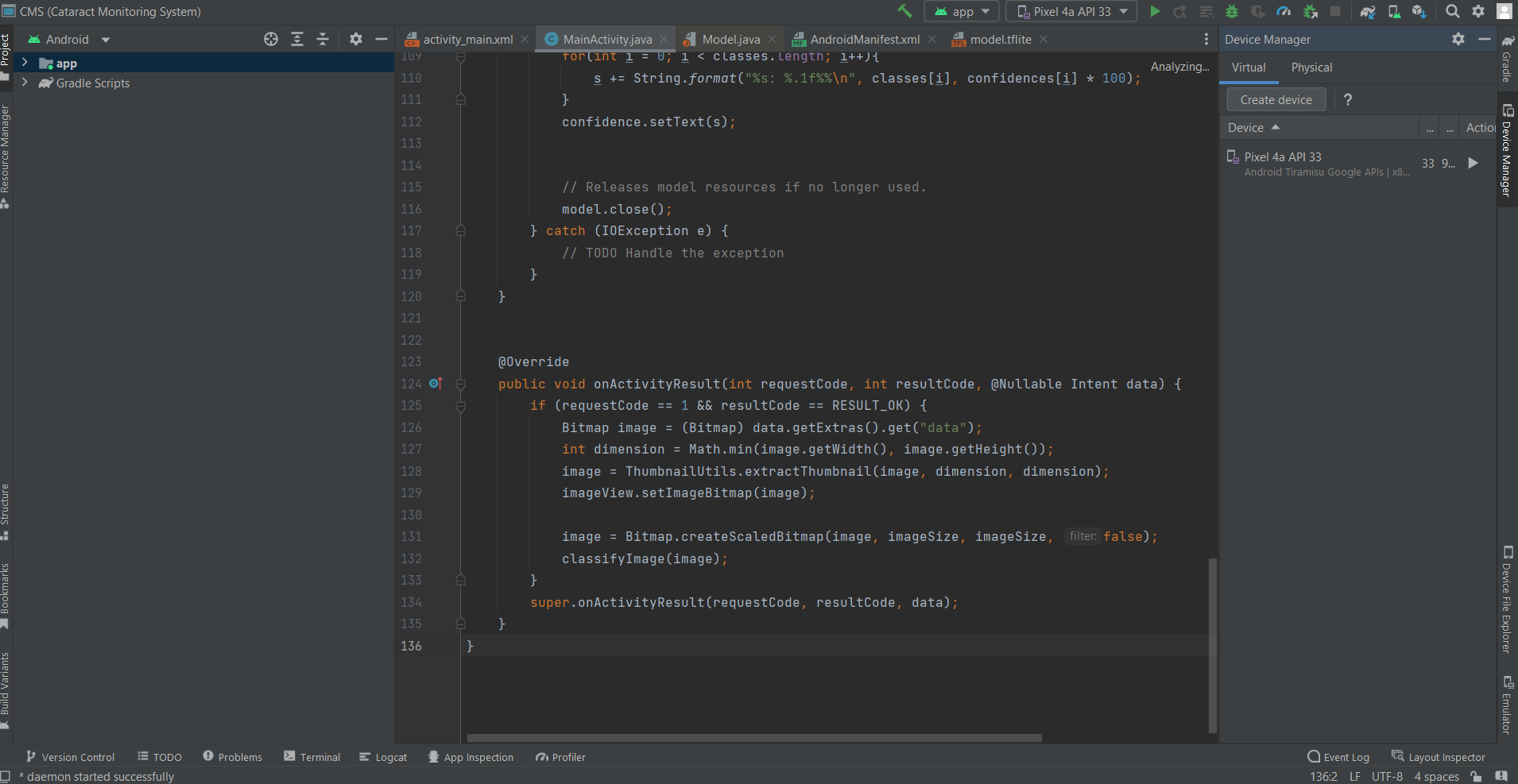
**Figure 3.3:** Anaconda Prompt

**JUPYTER NOTEBOOK:** It is basically a web application. Unlike IDEs, it uses the internet to run. And even after not being able to perform offline, it is highly preferred by most of the beginners because of its rich formatting and user-friendly interface. It allows us to enter the code in the browser and automatically highlights the syntax. It helps us to know if we are indenting the code correctly with the help of colors and bold formatting. For example, if we write the print command outside the scope of a loop, it will change the color of the print keyboard. Whitespace plays a very important role in Python because Python doesn’t involve the use of the braces for enclosing the bodies of loops, methods, etc. A single indentation mistake can lead to an error. The results are displayed in different representations like HTML, PNG, Latex, SVG, PDF, etc.



**Figure 3.4:** Jupyter Notebook

**Android Studio:** Android studio is an IDE integrated Development environment which is used for developing android applications



**Fig 3.5:** Android studio

## CHAPTER-4

**RESULTS AND OBSERVATIONS**

This chapter describes the results of our project and comparison of both the algorithms that we have used in our project to find the change between two timestamp images.

### 

### 4.1 Stepwise description of Results:

### In this project, we made use of Anaconda Prompt and Android Studio to identify that wheather a person is suffering form cataract or not

### A picture containing text Description automatically generatedText, letter Description automatically generated

### 

**Figure 4.1:** Test the eye **Figure 4.2:** Select the area of the eye and click classify

**4.2 Test Case Results:**

The above images are those which are obtained from the final analysis of our project.

* It will first checks wheather a person is suffering with cataract or not
* For that the person need to first open the app and click on the eye diseases so that he can test himself
* If the Person is suffering from cataract, The result says that he has “Cataract”. If “Normal” means he has “No cataract”

**CHAPTER-5: CONCLUSION AND FUTURE STUDY**

**5.1 CONCLUSION:**

In this project we have developed deeplearning models to identify cataract from the images of pupil of and eye we have developed claasifier model using cnn,vgg16 and inception models and we observed that the vgg16 model has got higher accuracy from the three models that we developed and we also develped an mobile application into which we integrated the deep learnig model that we developd

### 

### 5.2 Future Study:

The goal of the project is to develop an mobile application that can detect cataract we have achieved our basic goal.In future we can add feature like allowing users to upload their old photos and detect catarct from those photos and we also want to work on detecting other eye related diseases through smart phone

## REFERENCES

1. Askarian B, Ho P, Chong JW. Detecting cataract using smartphones. IEEE Journal of Translational Engineering in Health and Medicine. 2021 Apr 20;9:1-0.
2. Junayed MS, Islam MB, Sadeghzadeh A, Rahman S. CataractNet: An automated cataract detection system using deep learning for fundus images. IEEE Access. 2021 Sep 15;9:128799-808.
3. Hu S, Wang X, Wu H, Luan X, Qi P, Lin Y, He X, He W. Unified diagnosis framework for automated nuclear cataract grading based on smartphone slit-lamp images. IEEE Access. 2020 Sep 21;8:174169-78
4. Jindal I, Gupta P, Goyal A. Cataract detection using digital image processing. In2019 Global Conference for Advancement in Technology (GCAT) 2019 Oct 18 (pp. 1-4). IEEE.

## 

## APPENDIX

#importting the Keras Libraries and packages

from keras .models import Sequential

from keras.layers import Convolution2D

from keras.layers import MaxPooling2D

from keras.layers import Flatten

from keras.layers import Dense

from keras.utils.vis\_utils import plot\_model

#Initializing the CNN

classifier = Sequential()

#Convolution

classifier.add(Convolution2D(32,3,3, input\_shape=(64,64,3),activation='relu'))

#Pooling

classifier.add(MaxPooling2D(pool\_size=(2,2)))

#Addition of Second convolutional layers and pooling to get more accuracy on the test\_set

classifier.add(Convolution2D(32,3,3,activation='relu'))

classifier.add(MaxPooling2D(pool\_size=(2,2)))

#Flattening

classifier.add(Flatten())

#Full connection

classifier.add(Dense(128,activation='relu'))

classifier.add(Dense(1,activation='sigmoid'))

#Compiling the CNN

classifier.compile(optimizer='adam',loss='binary\_crossentropy',metrics=['accuracy'])

plot\_model(classifier, to\_file='model\_plot.png', show\_shapes=True, show\_layer\_names=True)

#Image preprocessing-fitting the CNN to image

from keras.preprocessing.image import ImageDataGenerator

train\_datagen = ImageDataGenerator(

rescale=1./255,

shear\_range=0.2,

zoom\_range=0.2,

horizontal\_flip=True)

test\_datagen = ImageDataGenerator(rescale=1./255)

training\_set = train\_datagen.flow\_from\_directory('C:/Users/Sukeerthi/Desktop/Dataset/Train',

target\_size=(64, 64),

batch\_size=9,

class\_mode='binary')

test\_set = test\_datagen.flow\_from\_directory('C:/Users/Sukeerthi/Desktop/Dataset/Test',

target\_size=(64, 64),

batch\_size=9,

class\_mode='binary')

history =classifier.fit(training\_set,

steps\_per\_epoch=575,

epochs=5,

validation\_data=test\_set,

validation\_steps=119)