A

Major Project

On

DRIVER DROWSINESS DETECTION SYSTEM

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

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

Dyagala Sanath Kumar
(18E41A0596)
Anumala Jithendra
(18E41A0584)
Kambham Rakesh Reddy
(18E41A0570)
Ponnala Veena
(18E41A0593)

Under the Guidance of

Mrs.C.Gouthami

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SREE DATTHA INSTITUTE OF ENGINEERING AND SCIENCE

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

Recognized Under Section 2(f) of the UGCAct.1956,

Sheriguda (V), Ibrahimpatnam (M), Ranga Reddy – 501510.

2018-22

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled "**DRIVER DROWSINESS DETECTION SYSTEM**" is being submitted by **D.SANATH KUMAR (18E41A0596), A.JITHENDRA (18E41A0584), K.RAKESH REDDY (18E41A0570), P.VEENA (18E41A0593)** 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 2018-22.

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. M. Varaprasad Rao **Dr. Md Sameeruddin Khan**

Internal Guide **Principal**

Dr. M. Varaprasad Rao

External Examiner

Hod

Submitted for viva-voice Examination held on ____________________

ACKNOWLEDGEMENT

Apart from our efforts, 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 for my guide **Dr. M. Varaprasad Rao,** Assistant 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 are also thankful to **Dr. M. Varaprasad Rao,** Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. Md. Sameeruddin Khan,** Principal for being cooperative throughout this project. We would like to express our sincere gratitude to **Dr. GNV Vibhav Reddy**, Vice-Chairman and **Sri. G. Panduranga Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout this project.

The guidance and support were received from all the members of **Sree Dattha Institute of Engineering and Science** 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.

ABSTRACT

This project is titled “**DRIVER DROWSINESS DETECTION SYSTEM**”. The main idea behind this project is to develop a nonintrusive system which can detect drowsiness of any human and can issue a timely warning. Drivers who do not take regular breaks when driving long distances run a high risk of becoming drowsy, a state which they often fail to recognize early enough. According to the expert’s studies show that around one quarter of all serious motorway accidents are attributable to sleepy drivers in need of a rest, meaning that drowsiness causes more road accidents than drink-driving. This system will monitor the driver's eyes using a camera and by developing an algorithm we can detect symptoms of driver fatigue early enough to avoid the person from sleeping. So, this project will be helpful in detecting driver fatigue in advance and will give warning output in the form of alarm and pop- ups.

Moreover, the warning will be deactivated manually rather than automatically. For this purpose, a de- activation dialog will be generated which will contain some simple mathematical operation which when answered correctly will dismiss the warning. Moreover, if driver feels drowsy there is a possibility of an incorrect response to the dialog. We can judge this by plotting a graph in the time domain. If all the three input variables show a possibility of fatigue at one moment, then a Warning signal is given in the form of text and sound. This will directly give an indication of drowsiness/fatigue which can be further used as a record of driver performance.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SCREENSHOTS

1. INTRODUCTION

1. INTRODUCTION

1.1 PROJECT SCOPE

Humans have always invented machines and devised techniques to ease and protect their lives, for mundane activities like traveling to work, or for more interesting purposes like aircraft travel. With the advancement in technology, modes of transportation kept on advancing and our dependence on it started increasing exponentially. It has greatly affected our lives as we know it. Now, we can travel to places at a pace that even our grandparents wouldn't have thought possible. In modern times, almost everyone in this world uses some sort of transportation every day. Some people are rich enough to have their own vehicles while others use public transportation. However, there are some rules and codes of conduct for those who drive irrespective of their social status. One of them is staying alert and active while driving.

1.2 PROJECT PURPOSE

Neglecting our duties towards safer travel has enabled hundreds of thousands of tragedies to get associated with this wonderful invention every year. It may seem like a trivial thing to most folks but following rules and regulations on the road is of utmost importance. While on road, an automobile wields the most power and in irresponsible hands, it can be destructive and sometimes, that carelessness can harm lives even of the people on the road. One kind of carelessness is not admitting when we are too tired to drive. In order to monitor and prevent a destructive outcome from such negligence, many researchers have written research papers on driver drowsiness detection systems. But at times, some of the points and observations made by the system are not accurate enough. Hence, to provide data and another perspective on the problem at hand, in order to improve their implementations and to further optimize the solution, this project has been done..

1.3 PROJECT FEATURES

The main features of this project are that the designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is a loop that ends as soon as the user is satisfied with the proposal.

2. SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

SYSTEM ANALYSIS

System Analysis is the 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 analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, “what must be done to solve the problem?” The system is viewed as a whole and the inputs to the system are identified. Once the analysis is completed the analyst has a firm understanding of what is to be done.

2.1 PROBLEM DEFINITION

A detailed study of the process must be made by various techniques like Image processing, feature recognition etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is a loop that ends as soon as the user is satisfied with the proposal.

2.2 EXISTING SYSTEM

By using non intrusive machine vision based concepts, drowsiness of the driver detected system is developed. Many existing systems require a camera which is installed in front of the driver . It points straight towards the face of the driver and monitors the driver’s eyes in order to identify the drowsiness. For large vehicles such as heavy trucks and buses this arrangement is not pertinent. Bus has a large front glass window to have a broad view for safe driving. If we place a camera on the window of the front glass, the camera blocks the frontal view of the driver so it is not practical. If the camera is placed on the frame which is just about the window, then the camera is unable to detain the anterior view of the face of the driver correctly. The open CV detector detects only 40% of the face of the driver in normal driving position in video recording of 10 minutes. In the oblique view, the OpenCV eye detector (CV-ED) frequently fails to trace the pair of eyes. If the eyes are closed for five successive frames the system concludes that the driver is slumbering and issues a warning signal [4]. Hence the existing system is not applicable for large vehicles. In order to conquer the problem of the existing system, a new detection system is developed in this project work.

2.2.1 LIMITATIONS OF EXISTING SYSTEM

- More classification.
- Time consuming.
- Needs manual calculations.

To avoid all these limitations and make the working more accurate the system needs to be implemented efficiently.

2.3 PROPOSED SYSTEM

The aim of the proposed system is to develop a system of improved facilities. The proposed system can overcome all the limitations of the existing system. The system provides higher accuracy and reduces the classification work. The existing system has several disadvantages and many more difficulties to work well. The proposed system tries to eliminate or reduce these difficulties up to some extent. The proposed system helps us to prevent accident accuracy.

2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM

The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations.

It has got following features:

- Minimum time needed for the various processing.
- Greater efficiency.
- Better service.

2.4 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and a business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of 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 effort is concentrated on a project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary 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 spend for the proposed system. Also all the resources are already available, it gives an indication that 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:

- 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 interfaces specify the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

• Processor	:	Intel Dual Core@ CPU 2.90GHz.
• Hard disk	:	16GB and Above.
• RAM	:	4GB and Above.
• Monitor	:	15 inches or above.

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 8, 10,11
• Languages	:	Python
• IDE	:	Spyder(anaconda),StarUML

3.ARCHITECTURE

3. ARCHITECTURE

3.1 PROJECT ARCHITECTURE

This project architecture shows the procedure followed for breed detection using machine learning, starting from input to final prediction.

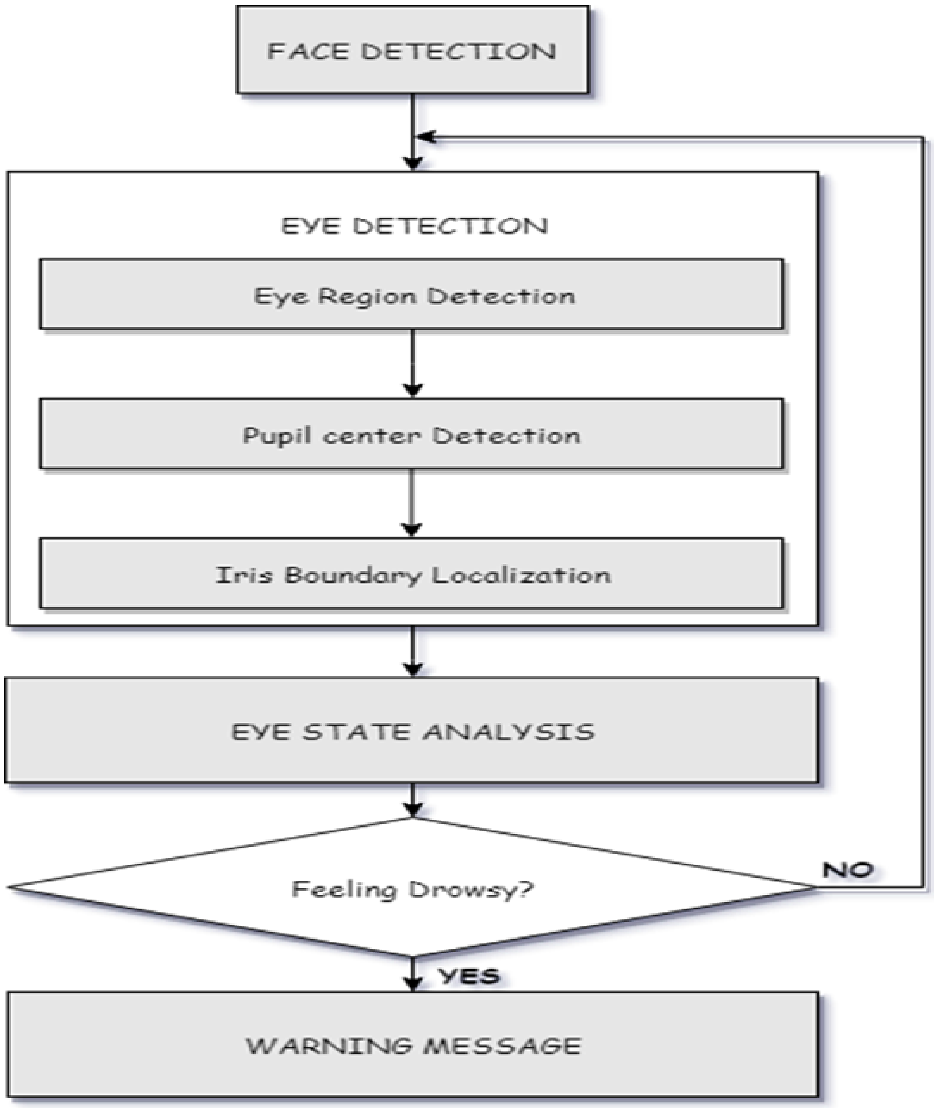


Figure 3.1: Project Architecture of DRIVER DROWSINESS DETECTION SYSTEM

3.2 USE CASE DIAGRAM

In the use case diagram we have basically two actors who are the user and the administrator. The user has the rights to login, access to resources and to view the crime details. Whereas the administrator has the login, access to resources of the users and also the right to update and remove the crime details, and he can also view the user files.

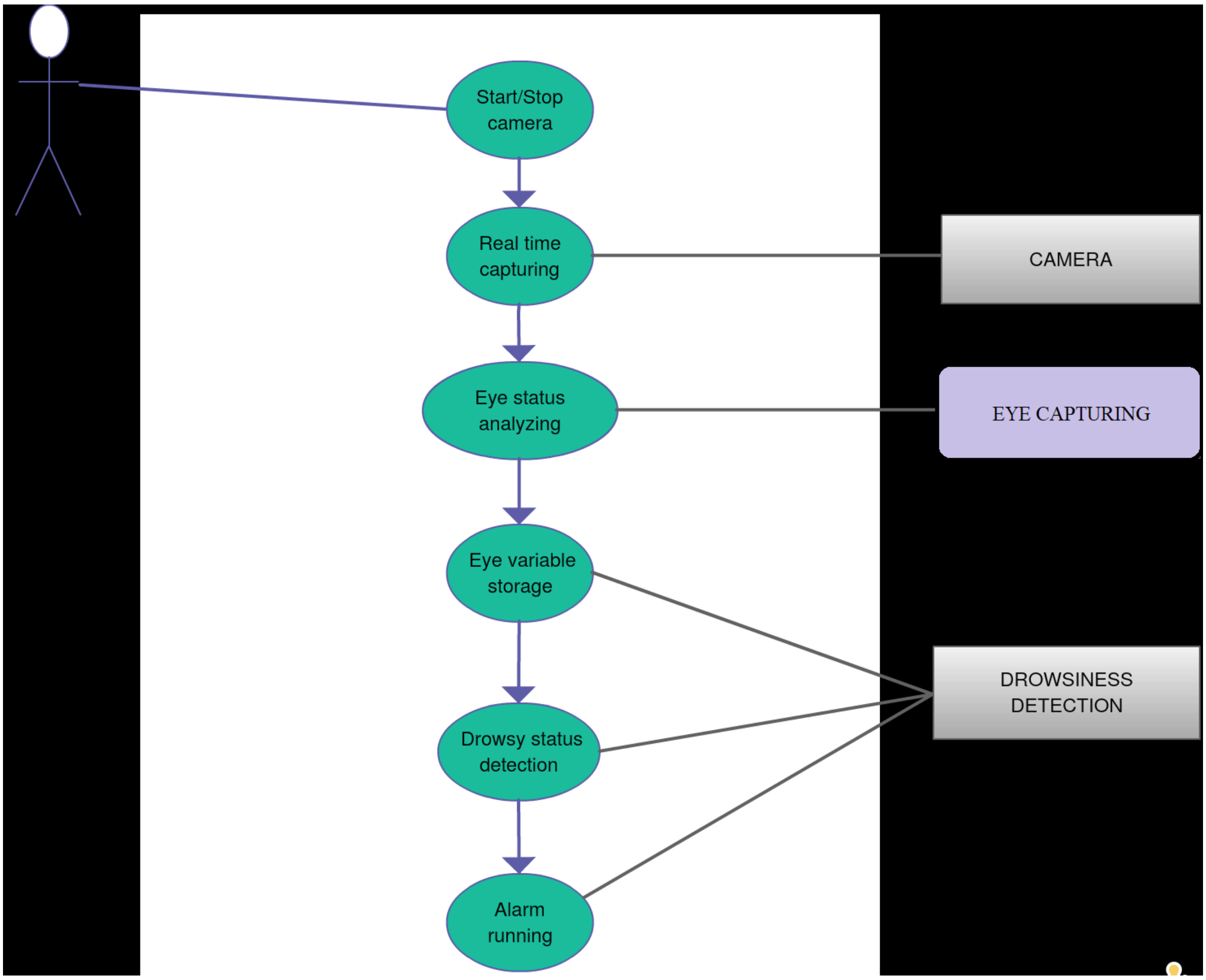
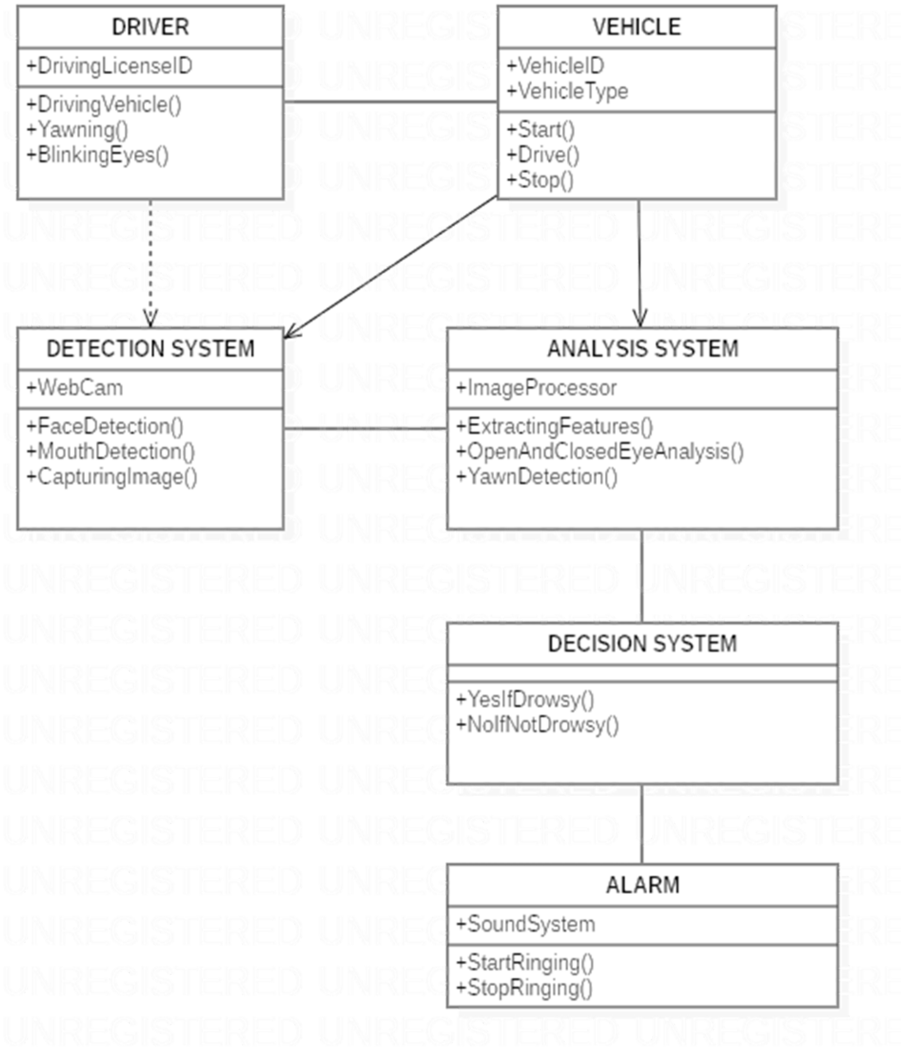


Figure 3.2: Use Case Diagram for user for DRIVER DROWSINESS DETECTION SYSTEM

3.3 CLASS DIAGRAM

Class Diagram is a collection of classes and objects.



UNREGISTERED

Figure 3.3: Class Diagram for Administrator and User for

DRIVER DROWSINESS DETECTION SYSTEM

3.4 SEQUENCE DIAGRAM

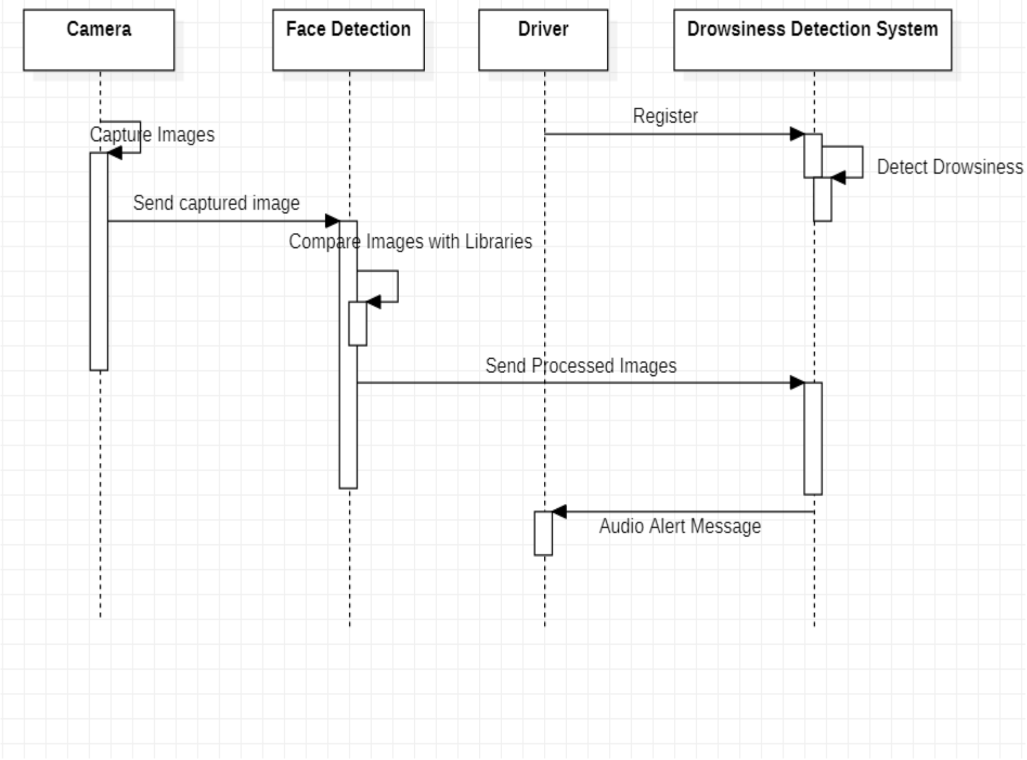


Figure 3.4: Sequence Diagram for DRIVER DROWSINESS DETECTION SYSTEM

3.5 ACTIVITY DIAGRAM

It describes the flow of activity states.

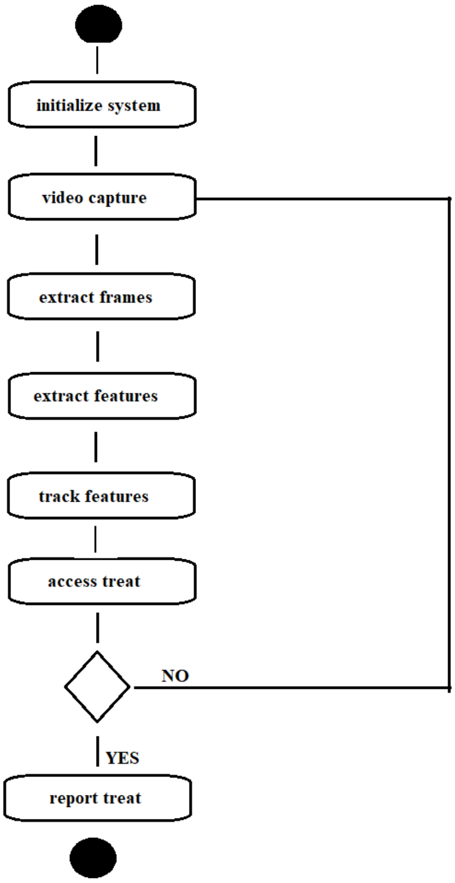


Figure 3.5: Activity Diagram for User for DRIVER DROWSINESS DETECTION SYSTEM

4. IMPLEMENTATION

4. IMPLEMENTATION

4.1 SAMPLE CODE

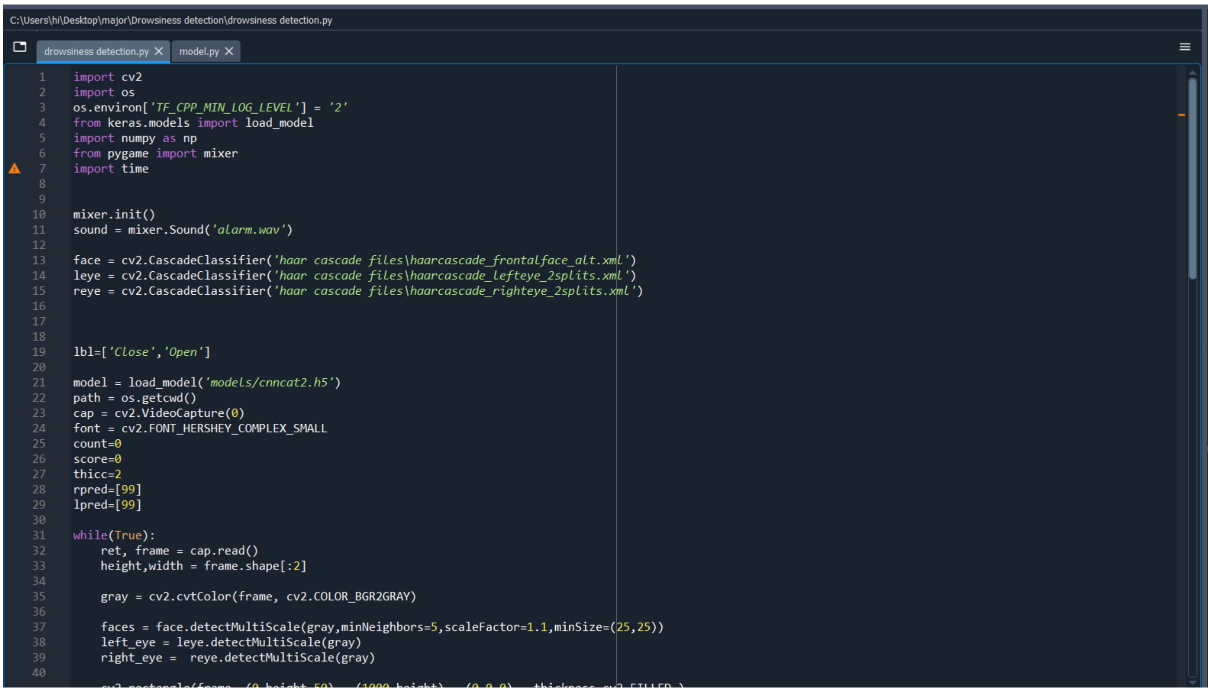
```
#drowsiness detection.py

import cv2
import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'
from keras.models import load_model
import numpy as np
from pygame import mixer
import time


mixer.init()
sound = mixer.Sound('alarm.wav')

face = cv2.CascadeClassifier('haar cascade files\haarcascade_frontalface_alt.xml')
leye = cv2.CascadeClassifier('haar cascade files\haarcascade_lefteye_2splits.xml')
reye = cv2.CascadeClassifier('haar cascade files\haarcascade_righteye_2splits.xml')



lbl=['Close','Open']

model = load_model('models/cnncat2.h5')
path = os.getcwd()
```

```
cap = cv2.VideoCapture(0)
font = cv2.FONT_HERSHEY_COMPLEX_SMALL
count=0
score=0
thicc=2
rpred=[99]
lpred=[99]

while(True):
    ret, frame = cap.read()
    height,width = frame.shape[:2]

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    faces = face.detectMultiScale(gray,minNeighbors=5,scaleFactor=1.1,minSize=(25,25))
    left_eye = leye.detectMultiScale(gray)
    right_eye =  reye.detectMultiScale(gray)

    cv2.rectangle(frame, (0,height-50) , (1000,height) , (0,0,0) , thickness=cv2.FILLED )

    for (x,y,w,h) in faces:
        cv2.rectangle(frame, (x,y) , (x+w,y+h) , (100,100,100) , 1 )

    for (x,y,w,h) in right_eye:
        r_eye=frame[y:y+h,x:x+w]
        count=count+1
        r_eye = cv2.cvtColor(r_eye,cv2.COLOR_BGR2GRAY)
        r_eye = cv2.resize(r_eye,(24,24))
        r_eye= r_eye/255
        r_eye=  r_eye.reshape(24,24,-1)
        r_eye = np.expand_dims(r_eye,axis=0)
        classes_x = model.predict(r_eye)
        rpred = np.argmax(classes_x,axis=1)
        if(rpred[0]==1):
            lbl='Open'
        if(rpred[0]==0):
            lbl='Closed'
        break

    for (x,y,w,h) in left_eye:
        l_eye=frame[y:y+h,x:x+w]
        count=count+1
        l_eye = cv2.cvtColor(l_eye,cv2.COLOR_BGR2GRAY)
        l_eye = cv2.resize(l_eye,(24,24))
        l_eye= l_eye/255
        l_eye=l_eye.reshape(24,24,-1)
        l_eye = np.expand_dims(l_eye,axis=0)
        classes_y = model.predict(l_eye)
        lpred = np.argmax(classes_y,axis=1)
        if(lpred[0]==1):
            lbl='Open'
        if(lpred[0]==0):
                lbl='Closed'
        break

    if(rpred[0]==0 and lpred[0]==0):
        score=score+1
        cv2.putText(frame,"Closed",(10,height-20), font, 1,(255,255,255),1,cv2.LINE_AA)
    # if(rpred[0]==1 or lpred[0]==1):
    else:
        score=score-2
        cv2.putText(frame,"Open",(10,height-20), font, 1,(255,255,255),1,cv2.LINE_AA)



    if(score<0):
        score=0
                    cv2.putText(frame,'Score:'+str(score),(100,height-20), font,
1,(255,255,255),1,cv2.LINE_AA)
    if(score>120):
        #person is feeling sleepy so we beep the alarm
        cv2.imwrite(os.path.join(path,'image.jpg'),frame)
                cv2.putText(frame,"Emergency indicators ON ",(250,height-10), font,
1,(255,255,255),1,cv2.LINE_AA)
        if(120<score<=150):
            cv2.putText(frame,"Speed Decrease to the limit 60Kmph",(250,height-10), font,
1,(255,255,255),1,cv2.LINE_AA)
        if(150<score<=170):
            cv2.putText(frame,"Speed Decrease to the limit 50Kmph",(250,height-10), font,
1,(255,255,255),1,cv2.LINE_AA)
        if(170<score<=200):
            cv2.putText(frame,"Speed Decrease to the limit 40Kmph",(250,height-10), font,
1,(255,255,255),1,cv2.LINE_AA)
        if(200<score):
                    cv2.putText(frame,"Engine turn off",(250,height-10), font,
1,(255,255,255),1,cv2.LINE_AA)
        try:
            sound.play()

        except:  # isplaying = False
            pass
        if(thicc<16):
            thicc= thicc+2
        else:
            thicc=thicc-2
            if(thicc<2):
                thicc=2
        cv2.rectangle(frame,(0,0),(width,height),(0,0,255),thicc)
    cv2.imshow('frame',frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()

 # Model.py


import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'
from keras.preprocessing import image
import matplotlib.pyplot as plt
import numpy as np
from keras.utils.np_utils import to_categorical
import random,shutil
from keras.models import Sequential
from keras.layers import Dropout,Conv2D,Flatten,Dense, MaxPooling2D, BatchNormalization
from keras.models import load_model



def generator(dir, gen=image.ImageDataGenerator(rescale=1./255), shuffle=True,batch_size=1,target_size=(24,24),class_mode='categorical' ):


return gen.flow_from_directory(dir,batch_size=batch_size,shuffle=shuffle,color_mode='grayscale',class_mode=class_mode,target_size=target_size)
```

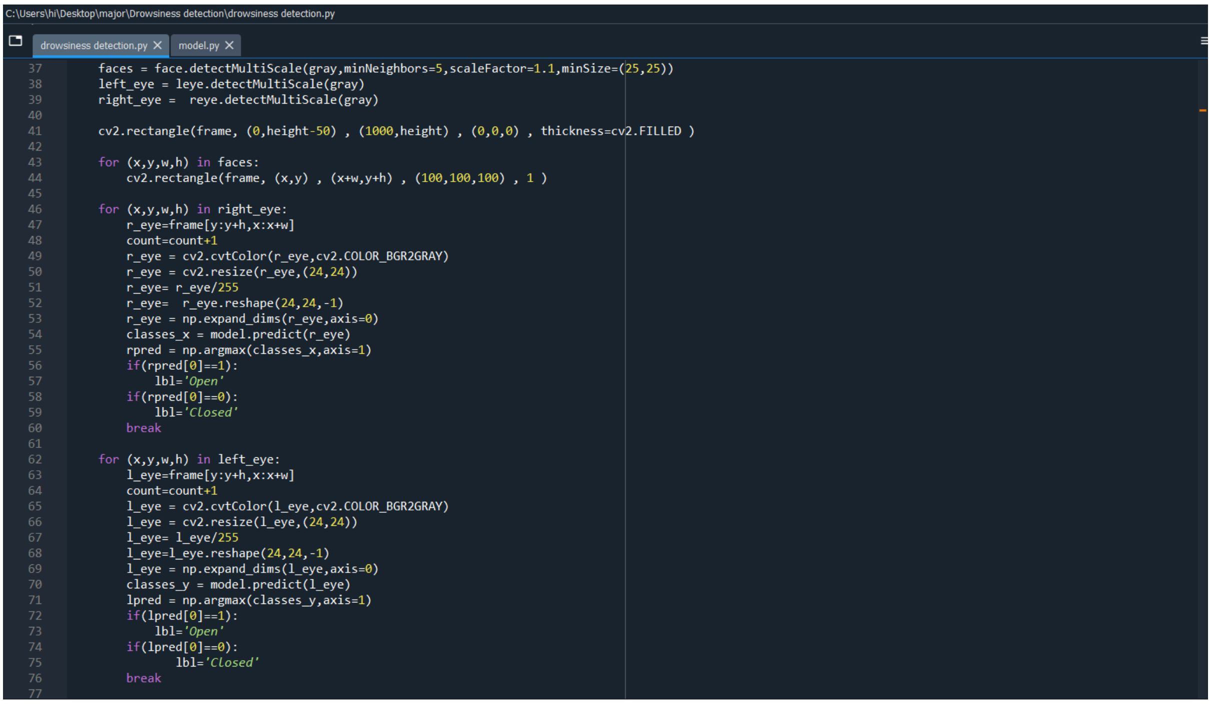
```
BS= 32
TS=(24,24)
train_batch= generator('train',shuffle=True, batch_size=BS,target_size=TS)
test_batch= generator('test',shuffle=True, batch_size=BS,target_size=TS)
SPE= len(train_batch.classes)//BS
VS = len(test_batch.classes)//BS
print(SPE,VS)


# img,labels= next(train_batch)
# print(img.shape)

model = Sequential([
    Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(24,24,1)),
    MaxPooling2D(pool_size=(1,1)),
    Conv2D(32,(3,3),activation='relu'),
    MaxPooling2D(pool_size=(1,1)),
#32 convolution filters used each of size 3x3
#again
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D(pool_size=(1,1)),

#64 convolution filters used each of size 3x3
#choose the best features via pooling

#randomly turn neurons on and off to improve convergence
    Dropout(0.25),
#flatten since too many dimensions, we only want a classification output
    Flatten(),
#fully connected to get all relevant data
   Dense(128, activation='relu'),
#one more dropout for convergence' sake :)
   Dropout(0.5),
#output a softmax to squash the matrix into output probabilities
   Dense(2, activation='softmax')
])

model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])

model.fit_generator(train_batch,
validation_data=test_batch,epochs=15,steps_per_epoch=SPE ,validation_steps=VS)

model.save('models/cnnCat2.h5', overwrite=True)
```

5. SCREENSHOTS

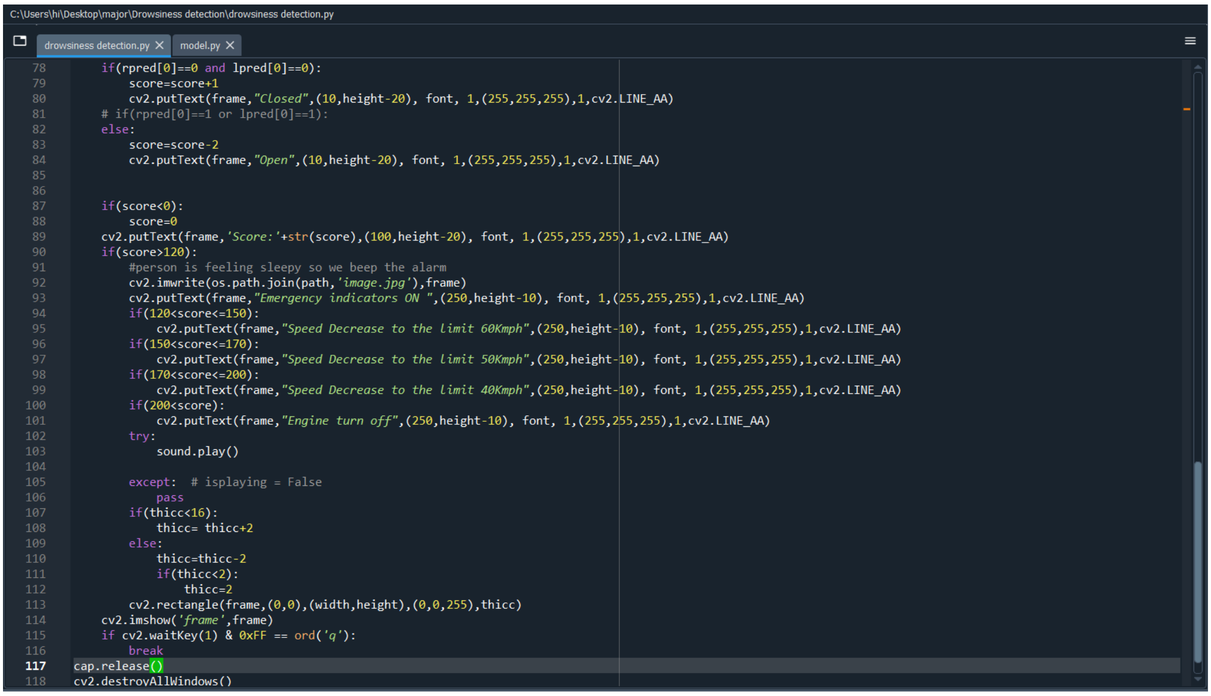
5.1 Drowsiness detection


C:\Users\hi\Desktop\major\Drowsiness detection\drowsiness detection.py
drowsiness detection.py
model.py

```
import cv2
import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'
from keras.models import load_model
import numpy as np
from pygame import mixer
import time


mixer.init()
sound = mixer.Sound('alarm.wav')

face = cv2.CascadeClassifier('haar cascade files\haarcascade_frontalface_alt.xml')
leye = cv2.CascadeClassifier('haar cascade files\haarcascade_lefteye_2splits.xml')
reye = cv2.CascadeClassifier('haar cascade files\haarcascade_righteye_2splits.xml')



lbl=['Close','Open']

model = load_model('models/cnncat2.h5')
path = os.getcwd()
cap = cv2.VideoCapture(0)
font = cv2.FONT_HERSHEY_COMPLEX_SMALL
count=0
score=0
thicc=2
rpred=[99]
lpred=[99]

while(True):
    ret, frame = cap.read()
    height,width = frame.shape[:2]

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    faces = face.detectMultiScale(gray,minNeighbors=5,scaleFactor=1.1,minSize=(25,25))
    left_eye = leye.detectMultiScale(gray)
    right_eye =  reye.detectMultiScale(gray)

    cv2.rectangle(frame, (0,height-50) , (1000,height) , (0,0,0) , thickness=cv2.FILLED )
```

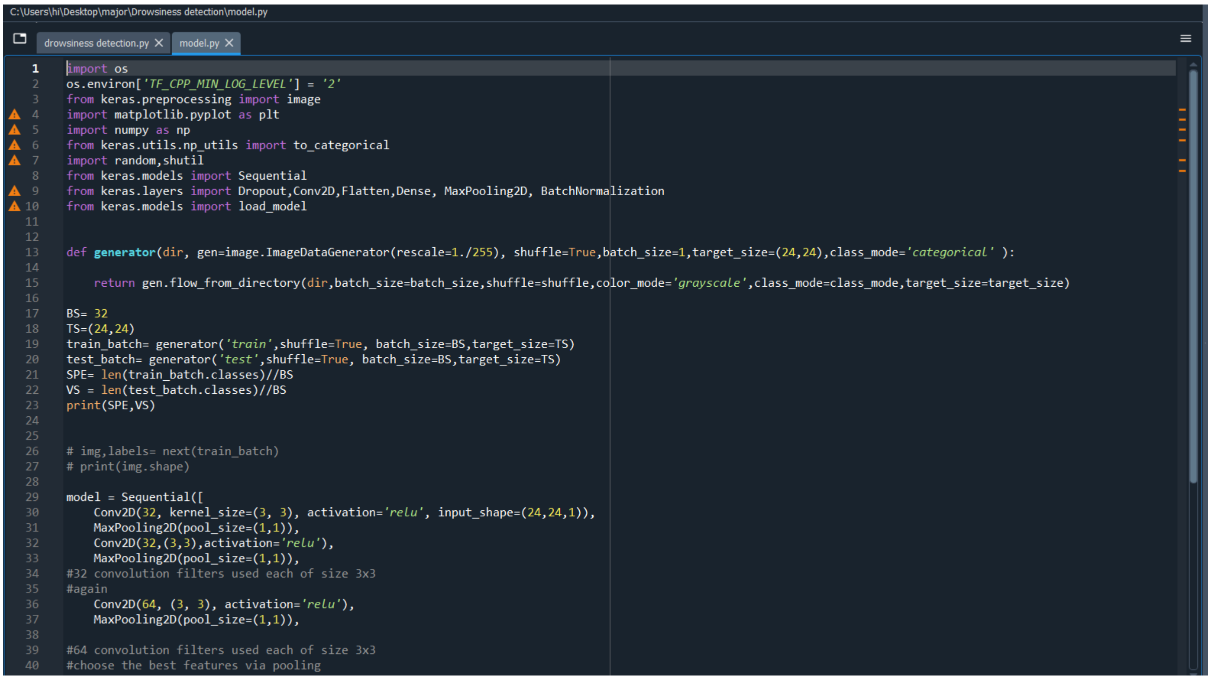

C:\Users\hi\Desktop\major\Drowsiness detection\drowsiness detection.py
drowsiness detection.py
model.py

```
    faces = face.detectMultiScale(gray,minNeighbors=5,scaleFactor=1.1,minSize=(25,25))
    left_eye = leye.detectMultiScale(gray)
    right_eye =  reye.detectMultiScale(gray)

    cv2.rectangle(frame, (0,height-50) , (1000,height) , (0,0,0) , thickness=cv2.FILLED )

    for (x,y,w,h) in faces:
        cv2.rectangle(frame, (x,y) , (x+w,y+h) , (100,100,100) , 1 )

    for (x,y,w,h) in right_eye:
        r_eye=frame[y:y+h,x:x+w]
        count=count+1
        r_eye = cv2.cvtColor(r_eye,cv2.COLOR_BGR2GRAY)
        r_eye = cv2.resize(r_eye,(24,24))
        r_eye= r_eye/255
        r_eye=  r_eye.reshape(24,24,-1)
        r_eye = np.expand_dims(r_eye,axis=0)
        classes_x = model.predict(r_eye)
        rpred = np.argmax(classes_x,axis=1)
        if(rpred[0]==1):
            lbl='Open'
        if(rpred[0]==0):
            lbl='Closed'
        break

    for (x,y,w,h) in left_eye:
        l_eye=frame[y:y+h,x:x+w]
        count=count+1
        l_eye = cv2.cvtColor(l_eye,cv2.COLOR_BGR2GRAY)
        l_eye = cv2.resize(l_eye,(24,24))
        l_eye= l_eye/255
        l_eye=l_eye.reshape(24,24,-1)
        l_eye = np.expand_dims(l_eye,axis=0)
        classes_y = model.predict(l_eye)
        lpred = np.argmax(classes_y,axis=1)
        if(lpred[0]==1):
            lbl='Open'
        if(lpred[0]==0):
               lbl='Closed'
        break
```

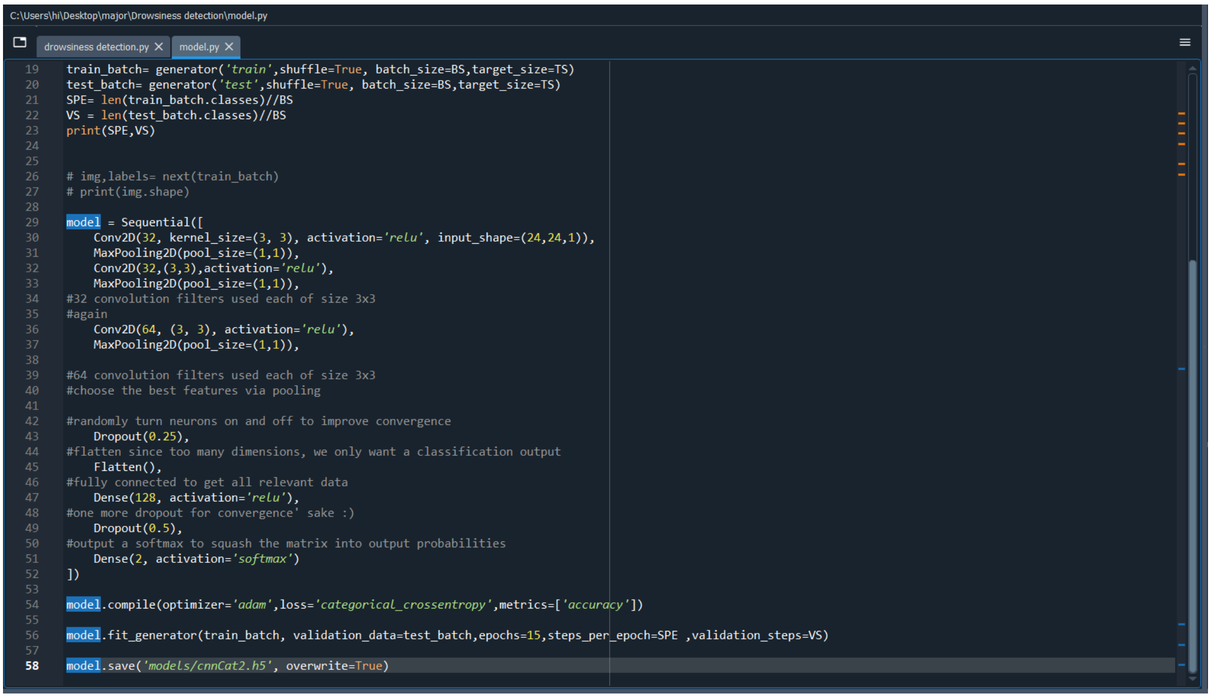

C:\Users\hi\Desktop\major\Drowsiness detection\drowsiness detection.py
drowsiness detection.py
model.py

```
    if(rpred[0]==0 and lpred[0]==0):
        score=score+1
        cv2.putText(frame,"Closed",(10,height-20), font, 1,(255,255,255),1,cv2.LINE_AA)
    # if(rpred[0]==1 or lpred[0]==1):
    else:
        score=score-2
        cv2.putText(frame,"Open",(10,height-20), font, 1,(255,255,255),1,cv2.LINE_AA)


    if(score<0):
        score=0
    cv2.putText(frame,'Score:'+str(score),(100,height-20), font, 1,(255,255,255),1,cv2.LINE_AA)
    if(score>120):
        #person is feeling sleepy so we beep the alarm
        cv2.imwrite(os.path.join(path,'image.jpg'),frame)
        cv2.putText(frame,"Emergency indicators ON ",(250,height-10), font, 1,(255,255,255),1,cv2.LINE_AA)
        if(120<score<=150):
            cv2.putText(frame,"Speed Decrease to the limit 60Kmph",(250,height-10), font, 1,(255,255,255),1,cv2.LINE_AA)
        if(150<score<=170):
            cv2.putText(frame,"Speed Decrease to the limit 50Kmph",(250,height-10), font, 1,(255,255,255),1,cv2.LINE_AA)
        if(170<score<=200):
            cv2.putText(frame,"Speed Decrease to the limit 40Kmph",(250,height-10), font, 1,(255,255,255),1,cv2.LINE_AA)
        if(200<score):
            cv2.putText(frame,"Engine turn off",(250,height-10), font, 1,(255,255,255),1,cv2.LINE_AA)
        try:
            sound.play()

        except:  # isplaying = False
            pass
        if(thicc<16):
            thicc= thicc+2
        else:
            thicc=thicc-2
            if(thicc<2):
                thicc=2
        cv2.rectangle(frame,(0,0),(width,height),(0,0,255),thicc)
    cv2.imshow('frame',frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

5.2 Model



```
import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'
from keras.preprocessing import image
import matplotlib.pyplot as plt
import numpy as np
from keras.utils.np_utils import to_categorical
import random,shutil
from keras.models import Sequential
from keras.layers import Dropout,Conv2D,Flatten,Dense, MaxPooling2D, BatchNormalization
from keras.models import load_model


def generator(dir, gen=image.ImageDataGenerator(rescale=1./255), shuffle=True,batch_size=1,target_size=(24,24),class_mode='categorical' ):

    return gen.flow_from_directory(dir,batch_size=batch_size,shuffle=shuffle,color_mode='grayscale',class_mode=class_mode,target_size=target_size)

BS= 32
TS=(24,24)
train_batch= generator('train',shuffle=True, batch_size=BS,target_size=TS)
test_batch= generator('test',shuffle=True, batch_size=BS,target_size=TS)
SPE= len(train_batch.classes)//BS
VS = len(test_batch.classes)//BS
print(SPE,VS)


# img,labels= next(train_batch)
# print(img.shape)

model = Sequential([
    Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(24,24,1)),
    MaxPooling2D(pool_size=(1,1)),
    Conv2D(32,(3,3),activation='relu'),
    MaxPooling2D(pool_size=(1,1)),
#32 convolution filters used each of size 3x3
#again
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D(pool_size=(1,1)),

#64 convolution filters used each of size 3x3
#choose the best features via pooling
```



```
train_batch= generator('train',shuffle=True, batch_size=BS,target_size=TS)
test_batch= generator('test',shuffle=True, batch_size=BS,target_size=TS)
SPE= len(train_batch.classes)//BS
VS = len(test_batch.classes)//BS
print(SPE,VS)


# img,labels= next(train_batch)
# print(img.shape)

model = Sequential([
    Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(24,24,1)),
    MaxPooling2D(pool_size=(1,1)),
    Conv2D(32,(3,3),activation='relu'),
    MaxPooling2D(pool_size=(1,1)),
#32 convolution filters used each of size 3x3
#again
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D(pool_size=(1,1)),

#64 convolution filters used each of size 3x3
#choose the best features via pooling
```

```

#randomly turn neurons on and off to improve convergence
    Dropout(0.25),
#flatten since too many dimensions, we only want a classification output
    Flatten(),
#fully connected to get all relevant data
    Dense(128, activation='relu'),
#one more dropout for convergence' sake :)
    Dropout(0.5),
#output a softmax to squash the matrix into output probabilities
    Dense(2, activation='softmax')
])

model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])

model.fit_generator(train_batch, validation_data=test_batch,epochs=15,steps_per_epoch=SPE ,validation_steps=VS)

model.save('models/cnnCat2.h5', overwrite=True)
```

6. TESTING

6. TESTING

6.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every 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 tests. 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 satisfactory, 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 centered 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.

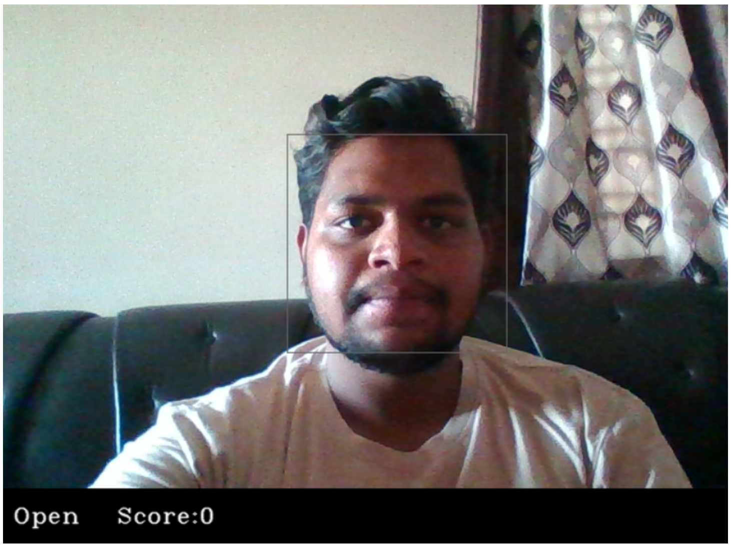
Output : identified classes of application outputs must be exercised.

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 identifying Business process flows; data fields, predefined processes.

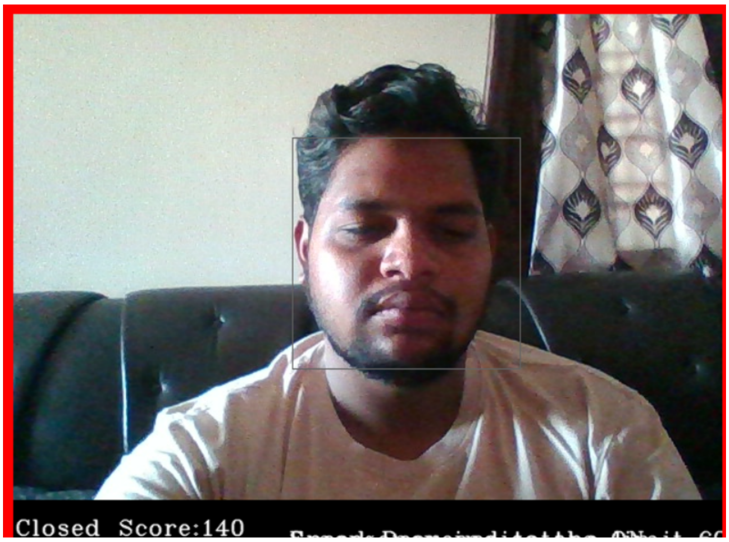
6.3 TEST CASES

6.31 UPLOADING IMAGES



```
Open   Score:0
```

NON - DROWSY PERSON



```
Closed  Score:140
```

DROWSY PERSON

7. CONCLUSION

7. CONCLUSION & FUTURE SCOPE

7.1 CONCLUSION

The project titled as "**DRIVER DROWSINESS DETECTION SYSTEM**" is a console based application. A program that localizes and tracks the vehicle driver's eyes and head movements developed to detect drowsiness. During monitoring, the proposed method decides whether eyes are open or closed, and whether the driver looks in front. An alert signal will be generated in the form of a buzzer or alarm when the device captures the movement of eyes closed for too long.

7.2 FUTURE SCOPE

The model can be improved incrementally by using other parameters like blink rate, yawning, speed of car, etc., If all these parameters are used it can improve the good rate of accuracy .We plan to further work on the project by adding a sensor to track the heart rate in order to prevent accidents caused due to sudden heart attacks to drivers.

Same model and techniques can be used for various other uses like Netflix and other streaming services can detect when the user is asleep and stop the video accordingly. It can also be used in applications that prevent users from sleeping.

8. BIBLIOGRAPHY

8. BIBLIOGRAPHY

8.1 REFERENCES

[1] COMPUTATIONALLY EFFICIENT FACE DETECTION; B. SCHLKOPF-A. BLAKE, S. ROMDHANI, AND P. TORR.

[2] USE OF THE HOUGH TRANSFORMATION TO DETECT LINES AND CURVES IN PICTURE; R. DUDA AND P. E. HART.

[3] JAIN, “FACE DETECTION IN COLOR IMAGES; R. L. HSU, M. ABDEL-MOTTALEB, AND A. K. JAIN.

[4] OPEN/CLOSED EYE ANALYSIS FOR DROWSINESS DETECTION; P.R. TABRIZI AND R. A. ZOROOFI.

[5] http://ncrb.gov.in/StatPublications/ADSI/ADSI2015/chapter1A%20traffic%20accidents.pdf

[6] http://www.jotr.in/text.asp?2013/6/1/1/118718

8.2 GitHub link