B. Tech (ECE)

Subject: Artificial Neural Networks (Semester: VII)

Assignment No: 02 (Group Activity)

Group Members:

Sr.No.	PRN No.	Name	
1.	1032190208	Shivani Kanyal	Leader
2.	1032190243	Kunal Singh	Member
3.	1032190269	Rishabh Sahay	Member
4.	1032191043	Sheerin Saxena	Member
5.	1032191122	Smit Arekar	Member

Objectives:

1. **To aid in the prevention of accidents passenger and commercial vehicles**.
2. The system will detect the early symptoms of drowsiness before the driver has fully lost all attentiveness and warn the driver that they are no longer capable of operating the vehicle safely.

Title: DRIVER DROWSINESS DETECTION SYSTEM USING CNN

Introduction:

Driver drowsiness and fatigue are significant causes of road accidents. Every year, they increase the number of deaths and fatalities worldwide. Driver fatigue affects the driving ability in the following 3 areas, a) It impairs coordination, b) It causes longer reaction times, and, c) It impairs judgment.
A module for an advanced driver assistance system is presented in this system to reduce the number of accidents caused by driver fatigue and thus increase transportation safety; this system deals with automatic driver drowsiness detection based on visual information and Artificial Intelligence.
The number of accidents as a result of drowsiness is increasing day by day. Recent statistics estimate that annually 76,000 injuries and 1200 deaths can be attributed to drowsiness related crashes. The advancement of technology in detecting the drowsiness of the driver is a noteworthy challenge as it can help reduce the probability of accidents↓↓ taking place resulting in decrease in the death and injuries caused due to drowsy driving .

Considering the hazards, drowsiness presents on the road, it is necessary to develop and efficient system which can work under low light environments and with better and faster speed. Nowadays driver safety in the car is one of the most wanted systems to avoid accidents.

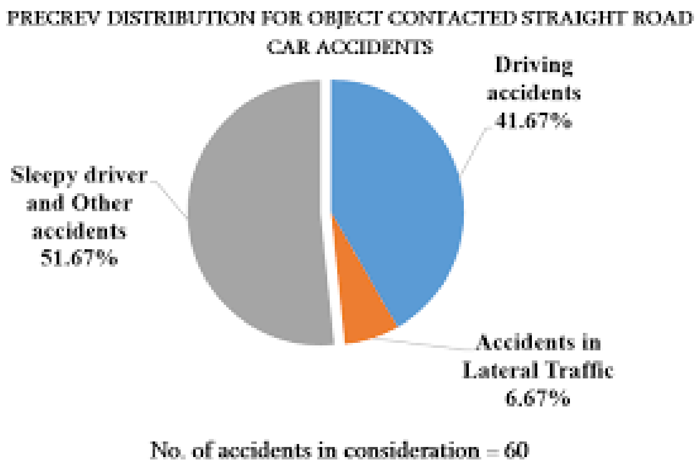
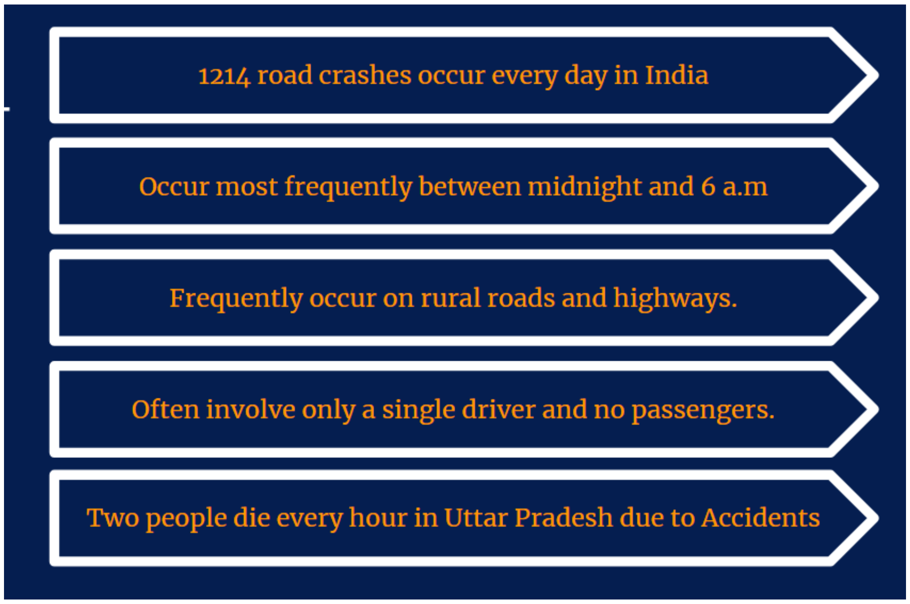


Fig 1.



1214 road crashes occur every day in India

Occur most frequently between midnight and 6 a.m

Frequently occur on rural roads and highways.

Often involve only a single driver and no passengers.

Two people die every hour in Uttar Pradesh due to Accidents

Fig 2.

Methodology:

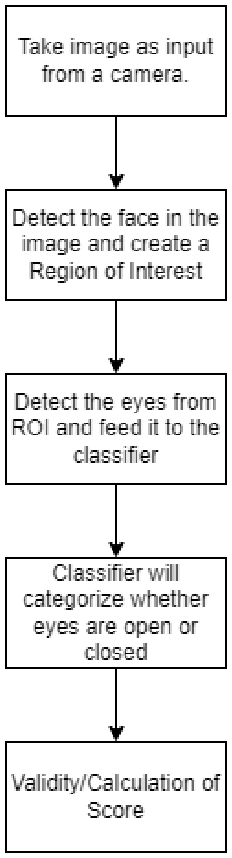

Take image as input from a camera.
Detect the face in the image and create a Region of Interest
Detect the eyes from ROI and feed it to the classifier
Classifier will categorize whether eyes are open or closed
Validity/Calculation of Score

Fig 3

Algorithm:

- Video captures both face and eyes.
- Then detection is used to extract both face and eyes regions.
- If the Driver is drowsy or not? IF the Eye Aspect Ratio has fallen below the threshold?
- If the result has fallen below the threshold level then the system sends the signal to the alarm.
- The alarm buzzes and alerts the driver that he is sleepy and must take a break.
- If the tracking is good that is the Eye Aspect Ratio is normal then we return to the initialization step

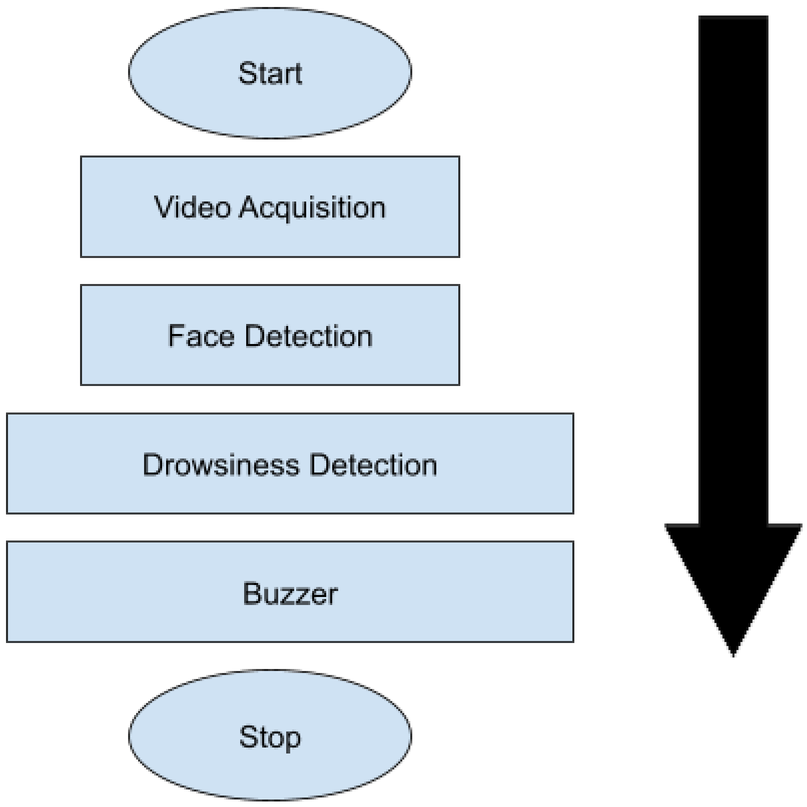


Fig 4

The system begins with the initialization phase, which is video acquisition of both face and eyes. Then detection is used to extract both face and eye regions and take them as frames to track them in real time.

For each tracking we test if that tracking is good or bad? If the Driver is drowsy or not? IF the Eye Aspect Ratio has fallen below the threshold? If the result has fallen below the threshold level then the system sends the signal to the alarm. The alarm buzzes and alerts the driver that he is sleepy and must take a break. If the tracking is good, that is the Eye Aspect Ratio is normal then we return to the initialization step and continue to take the next frames from the video.

The system will try to track the face again and again if no face is detected. The loud alarm has many positive effects as not only the driver but the passengers seated in the car may also get aware that the driver is sleepy and can take the required actions.

Working of System:

1. Power on the system.

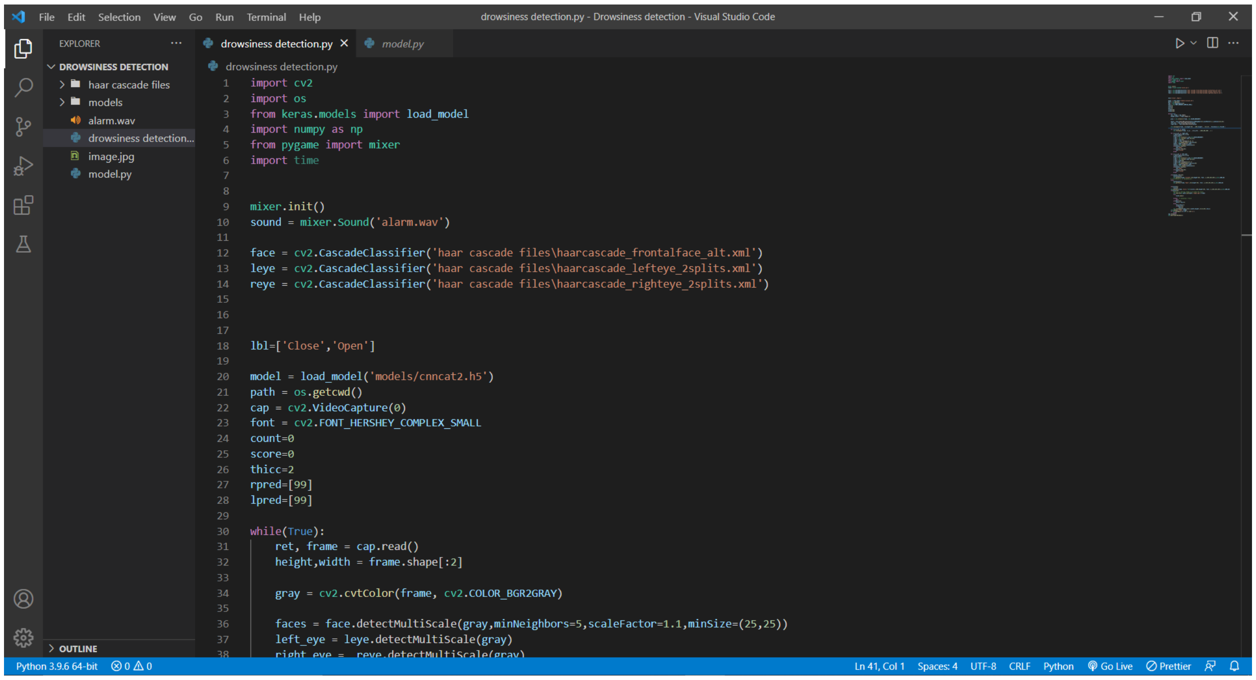
2. Camera will capture the video of the driver.

3. Raspberry Pi will work on the frames of the video.

4. If the system detects the drowsiness of the driver then the buzzer will beep.

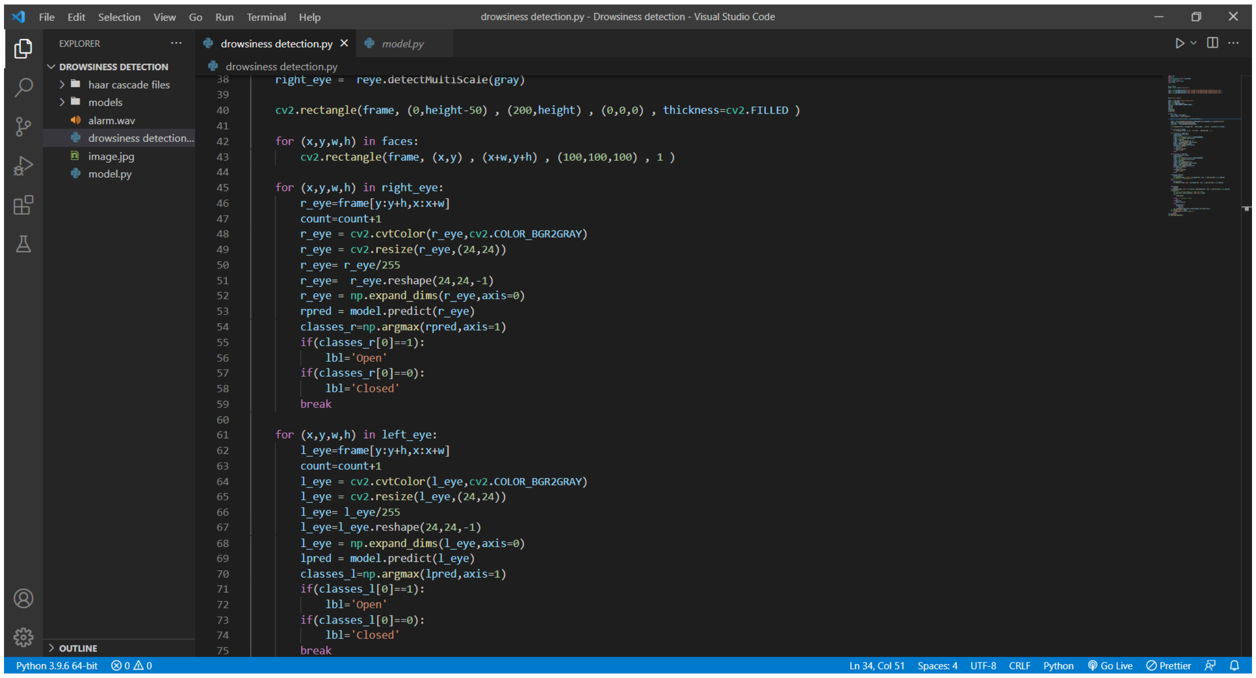
Implementation

1. drowsiness_detection.py



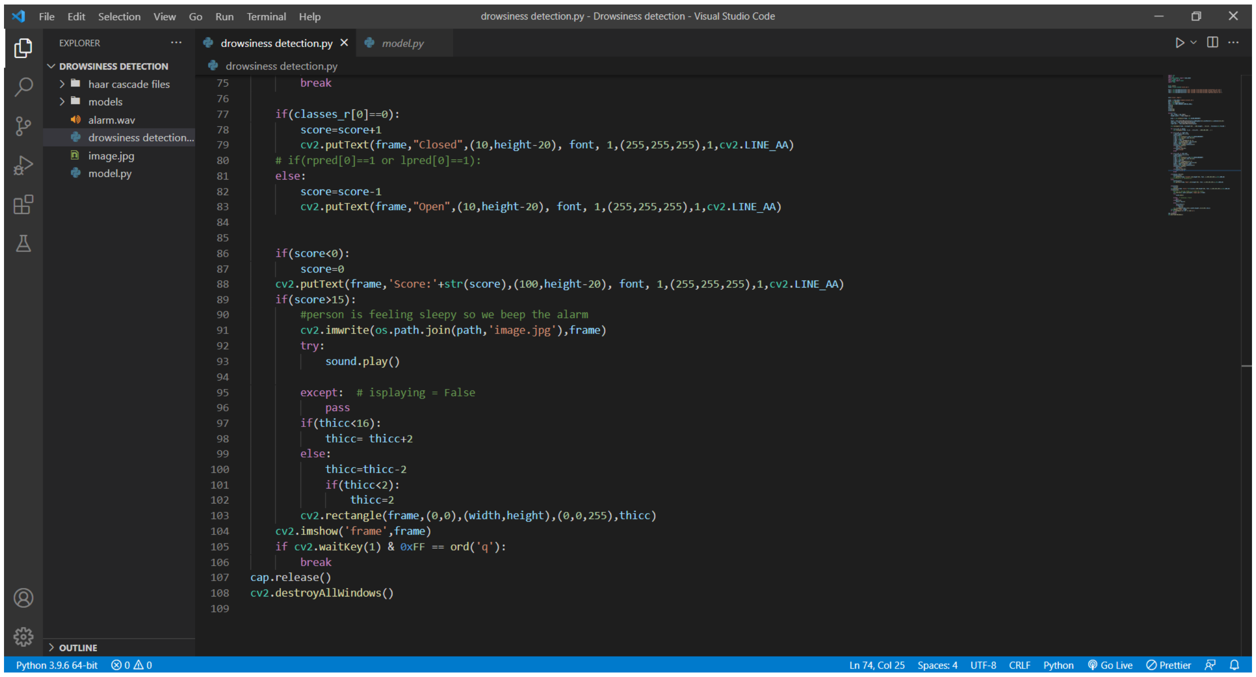
```python
import cv2
import os
from keras.models import load_model
import numpy as np
from pygame import mixer
import time


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)
```

Fig 5



```python
    right_eye =  reye.detectMultiScale(gray)

    cv2.rectangle(frame, (0,height-50) , (200,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)
        rpred = model.predict(r_eye)
        classes_r=np.argmax(rpred,axis=1)
        if(classes_r[0]==1):
            lbl='Open'
        if(classes_r[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)
        lpred = model.predict(l_eye)
        classes_l=np.argmax(lpred,axis=1)
        if(classes_l[0]==1):
            lbl='Open'
        if(classes_l[0]==0):
            lbl='Closed'
        break
```

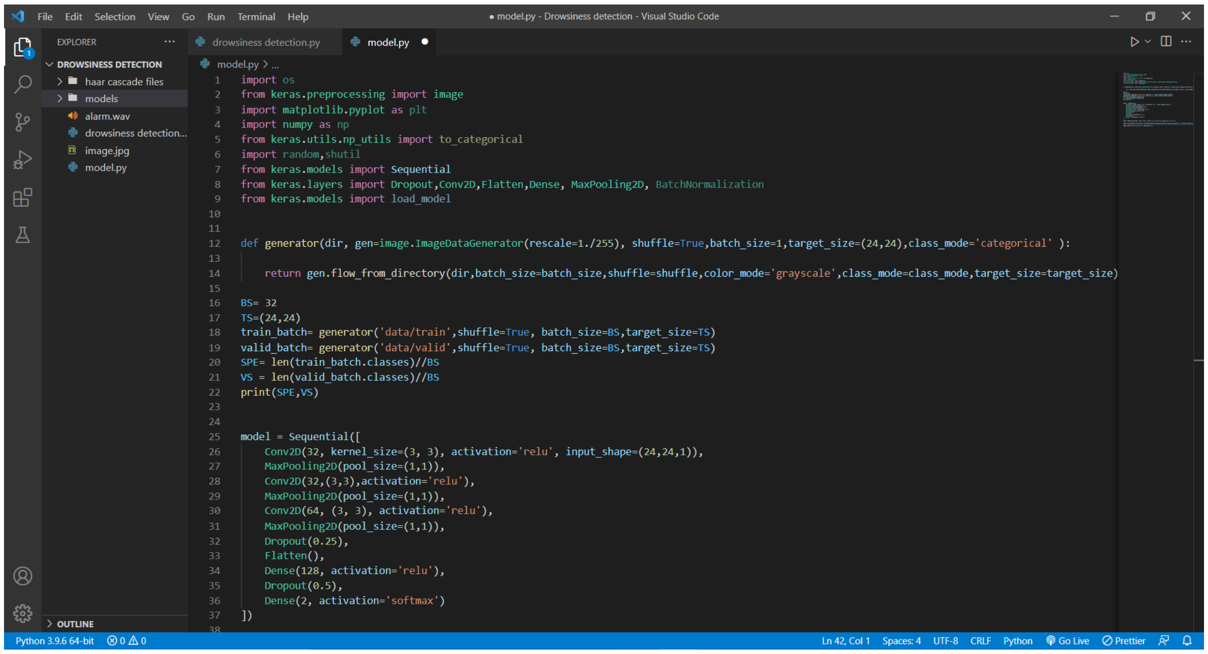
Fig 6



```python
            break

        if(classes_r[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-1
            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>15):
            #person is feeling sleepy so we beep the alarm
            cv2.imwrite(os.path.join(path,'image.jpg'),frame)
            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()
```

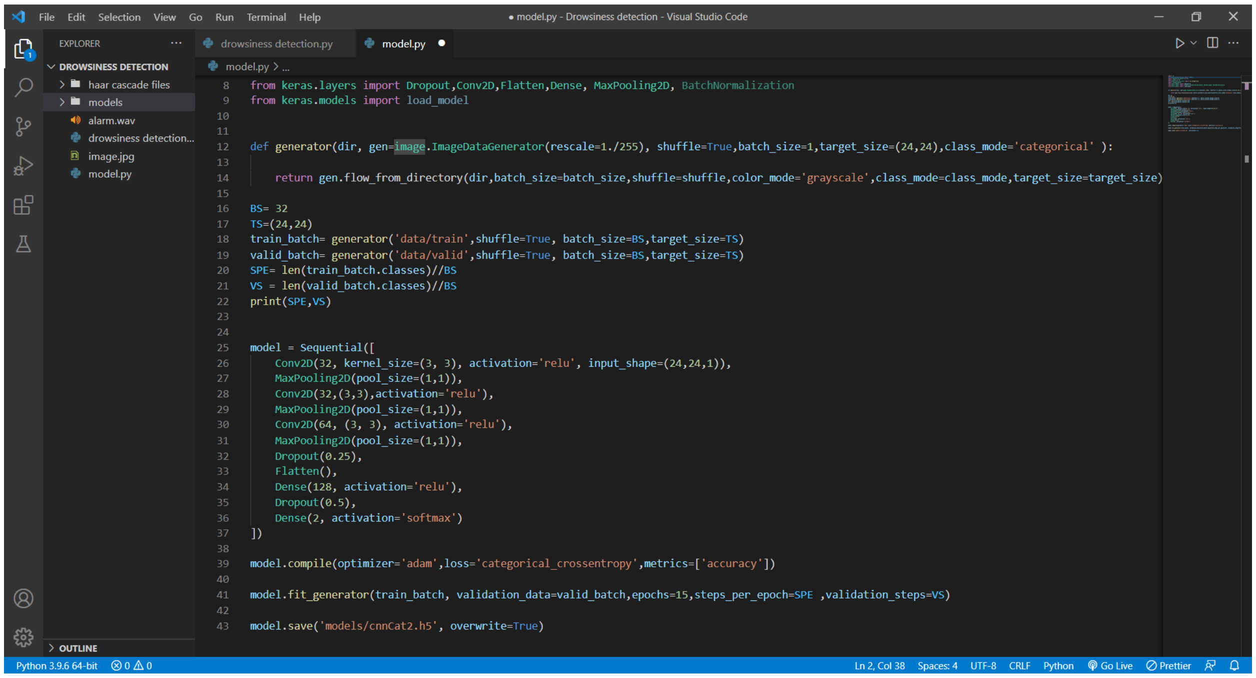
Fig 7

2. model.py



```python
import os
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('data/train',shuffle=True, batch_size=BS,target_size=TS)
valid_batch= generator('data/valid',shuffle=True, batch_size=BS,target_size=TS)
SPE= len(train_batch.classes)//BS
VS = len(valid_batch.classes)//BS
print(SPE,VS)


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)),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D(pool_size=(1,1)),
    Dropout(0.25),
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(2, activation='softmax')
])
```

fig 4.4



```python
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('data/train',shuffle=True, batch_size=BS,target_size=TS)
valid_batch= generator('data/valid',shuffle=True, batch_size=BS,target_size=TS)
SPE= len(train_batch.classes)//BS
VS = len(valid_batch.classes)//BS
print(SPE,VS)


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)),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D(pool_size=(1,1)),
    Dropout(0.25),
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(2, activation='softmax')
])
```

```python

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

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

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

Fig 4.5

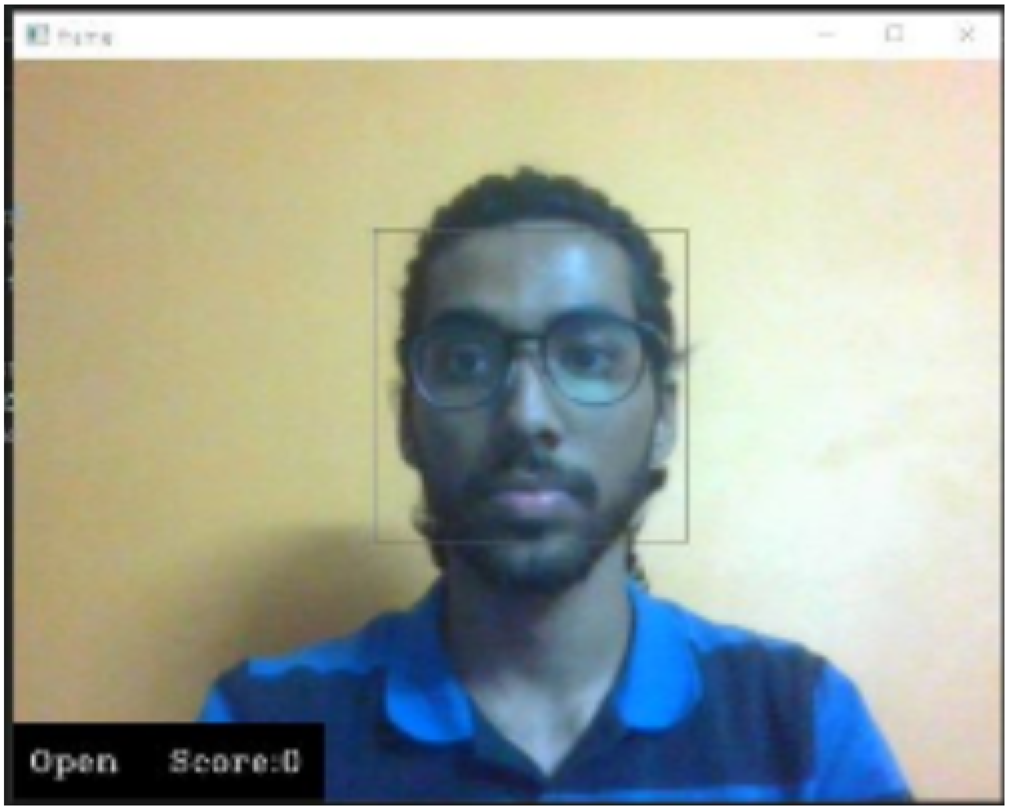


Fig 4.6 ‘Awake’

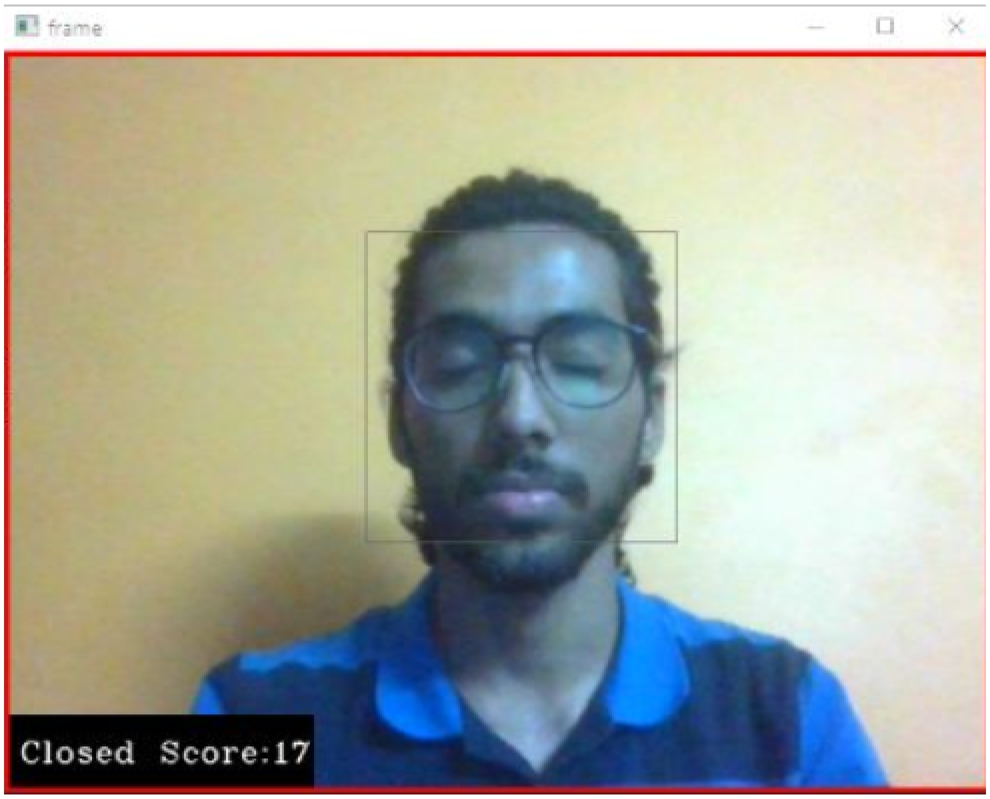


Fig 4.7 - ‘Asleep’

5. RESULTS

The drowsiness detection and correction system developed is capable of detecting drowsiness in a rapid manner. The system which can differentiate normal eye blink and drowsiness which can prevent the driver from entering the state of sleepiness while driving. The system works well even in case of drivers wearing spectacles and under low light conditions also. The alarm goes on whenever the system detects that the driver is asleep. The system works accurately for almost all conditions and is efficient too.

6. CONCLUSION

Drowsy driving is a serious threat to drivers and traffic participants. The present's system lacks one or the other important feature that provides non-reliable results. Our proposed system will overcome these drawbacks and provide accurate and reliable results.

The general flow of our drowsiness detection algorithm is fairly straightforward. A camera is set up to monitor the stream of faces. After which, we apply facial landmark detection and extract the eye regions. We can compute the eye aspect ratio to determine if the eyes are closed. Video segments whose average eye state point exceeds the threshold value are detected as drowsy and the driver is alerted. The system can also be used efficiently in locomotives and aero planes. During the monitoring, the system is able to decide if the eyes are opened or closed. When the eyes have been closed for about two seconds, the alarm beeps to alert the driver and the speed of the vehicle is reduced.

By doing this many accidents will be reduced and provide safe life to the driver and vehicle safety. A system for driver safety and car security is presented only in the luxurious costly cars. Using drowsiness detection system, driver safety can be implemented in normal cars also

1. REFERENCES

1. Therdpong Daengsi, Teerawat Poonwichein, Pongpisit Wuttidittachotti, "Drowsiness/Sleep Detection and Alert System: Development of A

Prototype Using Raspberry PI", Emerging Technology (INCET)2021 2nd International Conference for, pp. 1-4, 2021

2. Sanjay S, N. Banupriya, Sathish M, Sujay Nithish H, "Drowsiness Detection with OpenCV", Electronics and Sustainable Communication Systems (ICESC) 2021 Second International Conference on, pp. 1421-1425, 2021.
3. Bagus G. Pratama, IgiArdiyanto, Teguh B. Adji, A Review on Driver Drowsiness Based on Image, Bio-Signal, and Driver Behavior, IEEE, July 2017.
4. https://towardsdatascience.com/drowsiness-detection-with-machine-learning -765a16ca208a