Project Report on

Driver Drowsiness Detection System

Mini Project 1B of Second Year,

(SE Semester-IV) By

Shraddheya Kadam

B-63 TU2S2021007

Suraj Mane

B-87 TU2S2021031

Omkar Thakare

B- 86 TU2S2021030

Amol Gole

B- 65 TU2S2021009

Under the guidance of

Prof. Deepak Kurule

Department of Electronics and Telecommunication Engineering

TERNA ENGINEERING COLLEGE

Plot no.12, Sector-22, Opp. Nerul Railway station,
Phase-11, Nerul (w), Navi Mumbai 400706

UNIVERSITY OF MUMBAI

TERNA ENGINEERING COLLEGE

Plot no.12, Sector-22, Opp. Nerul Railway station,

Phase-11, Nerul (w), Navi Mumbai 400706

CERTIFICATE

Driver Drowsiness Detection

Submitted by:

Shraddheya Kadam	Suraj Mane
B-63 TU2S2021007	B-87 TU2S2021031
Omkar Thakare	Amol Gole
B- 86 TU2S2021030	B-65 TU2S2021009

in fulfilment of the

Mini Project 1B of Second Year,

(SE Sem-IV) in Electronics & Telecommunication Engineering.

Name of Guide	**Principal**	**Head of Department**
Prof. Deepak Kurule	Dr.. L.K.Ragha.	Dr. Jyoti Digge.

Approval Sheet

Project Report Approval

This Mini Project Report – titled “**Driver Drowsiness Detection System using Raspberry Pi**” by following students is approved for the degree of **B.E. in “Electronics & Telecomm. Engineering”**

Submitted by:

Shraddheya Kadam	B-63 TU2S2021007
Suraj Mane	B-87 TU2S2021031
Omkar Thakare	B- 86 TU2S2021030
Amol Gole	B- 65 TU2S2021009

Examiner’s Name & Signature:

1. ____________________

2. ____________________

Date: ___________________

Place: ___________________

Declaration

We declare that this written submission represents our ideas in my own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Shraddheya Kadam	B-63 TU2S2021007	__________
Suraj Mane	B-87 TU2S2021031	__________
Omkar Thakare	B- 86 TU2S2021030	__________
Amol Gole	B- 65 TU2S2021009	__________

Acknowledgement

We would like to express our sincere gratitude towards our guide **Prof. Deepak Kurule,** for their help, guidance and encouragement, they provided during the project development. This work would have not been possible without their valuable time, patience and motivation. We thank them for making our stint thoroughly pleasant and enriching. It was great learning and an honor being their student.

We are deeply thankful to **Dr. Jyothi Digge (H.O.D EXTC Department)** and entire team in the EXTC Department. They supported us with scientific guidance, advice and encouragement, they were always helpful and enthusiastic and this inspired us in our work.

We take the privilege to express our sincere thanks **to Dr. L. K. Ragha** our Principal for providing the encouragement and much support throughout our work.

Shraddheya Kadam	B-63 TU2S2021007	__________
Suraj Mane	B-87 TU2S2021031	__________
Omkar Thakare	B- 86 TU2S2021030	__________
Amol Gole	B- 65 TU2S2021009	__________

Date: ______________________

Place: ______________________

ABSTRACT

This document is a review report on the research conducted and the project made to develop a system for driver drowsiness detection to prevent accidents from happening because of driver fatigue and sleepiness. The report proposed the results and solutions on the limited implementation of the various techniques that are introduced in the project. Whereas the implementation of the project gives the real world idea of how the system works and what changes can be done in order to improve the utility of the overall system.

Furthermore, the report states the overview of the observations made by the authors in order to help further optimization in the mentioned field to achieve the utility at a better efficiency for a safer road.

Keywords—Driver drowsiness; eye detection; yawn detection; blink pattern; fatigue

INDEX

1. INTRODUCTION

- **Driver drowsiness detection** is a car safety technology which helps prevent accidents caused by the driver getting drowsy.
- Various studies have suggested that around 20% of all road accidents are fatigue-related, up to 50% on certain roads.
- The driver drowsiness detection is based on an algorithm, which begins recording the driver's steering behavior the moment the trip begins. It then recognizes changes over the course of long trips, and thus also the driver's level of fatigue.
- Typical signs of waning concentration are phases during which the driver is barely steering, combined with slight, yet quick and abrupt steering movements to keep the car on track.
- Based on the frequency of these movements and other parameters, use of turn signals, and the time of day, the function calculates the driver's level of fatigue. If that level exceeds a certain value, then this system warns drivers that they need a rest.

1.1 Need :-

- Sleep deprived drivers remain responsible for about 40% of the road accidents, according to enforcement officers patrolling the highways and major roads.(Sours The HIndu).
- According to the National Sleep Foundation(US), about half of U.S. adult drivers admit to consistently getting behind the wheel while feeling drowsy. About 20% admit to falling asleep behind the wheel at some point in the past year – with more than 45% admitting this has happened at least once in their driving careers.
- These startling figures show how prevalent drowsy driving is. What drivers may not realize is how much drowsy driving puts themselves and others at risk. In fact, an estimated 5,000 people died in 2015 in crashes involving drowsy driving, according to a Governors Highway Safety Association report
- A study by the AAA Foundation for Traffic Safety estimated that 328,000 drowsy driving crashes occur annually. That's more than three times the police-reported number. The same study found that 109,000 of those drowsy driving crashes resulted

in an injury and about 6,400 were fatal. The researchers suggest the prevalence of drowsy driving fatalities is more than 350% greater than reported.

- This type of project helps us to overcome about 20-30 percent of accidents and save more life as possible.

1.2 Scope :-

- There are many products out there that provide the measure of fatigue level in the drivers which are implemented in many vehicles.
- The driver drowsiness detection system provides similar functionality but with better results and additional benefits. Also, it alerts the user on reaching a certain saturation point of the drowsiness measure.

1.3 Problem Statement:-

- Designing a prototype Drowsiness Detection System which will focus on continuously.
- Accurately monitoring the state of the driver's eyes in real time to check whether they are open or closed for more than a given period of time

2. TECHNOLOGY USED

Python :-

- Python is an interpreted, high-level, general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace.
- Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed AND supports multiple programming paradigms, including procedural, object-oriented, and functional programming.

Image Processing :-

- In computer science, digital image processing is the use of computer algorithms to perform image processing on digital images.

2.1 Software requirements specification :-

Python :

- Python 3

Libraries :

- Numpy
- SciPy
- Time
- Argparse
- Dlib
- Imutils
- opencv
- Thread, etc..

Operating System :

- Raspbian os 5.10

Hardware Requirements Specification :

- Raspberry Pi 4
- Raspberry Pi Camera Module V1
- Laptop

2.2 Requirement analysis :-

Python:

- Python is the basis of the program that we wrote. It utilizes many of the python libraries.

Libraries:

- **Numpy:** Pre-requirement for Dlib
- **Scipy:** Used for calculating Euclidean distance between the eyelids.
- **Playsound:** Used for sounding the alarm
- **Dlib:** This program is used to find the frontal human face and estimate its pose using 68 face landmarks.
- **Imutils:** Convenient functions written for Opencv.
- **Opencv:** Used to get the video stream from the webcam, etc.

OS: Program is tested on Raspbian os 5.10

Raspberry Pi 4: Used to run our code.

Raspberry pi Camera: Used to get video feed.

Laptop:- Using VNC laptop used as monitor for Raspberry pi 4.

3. SYSTEM DEVELOPMENT

3.1 Methodology

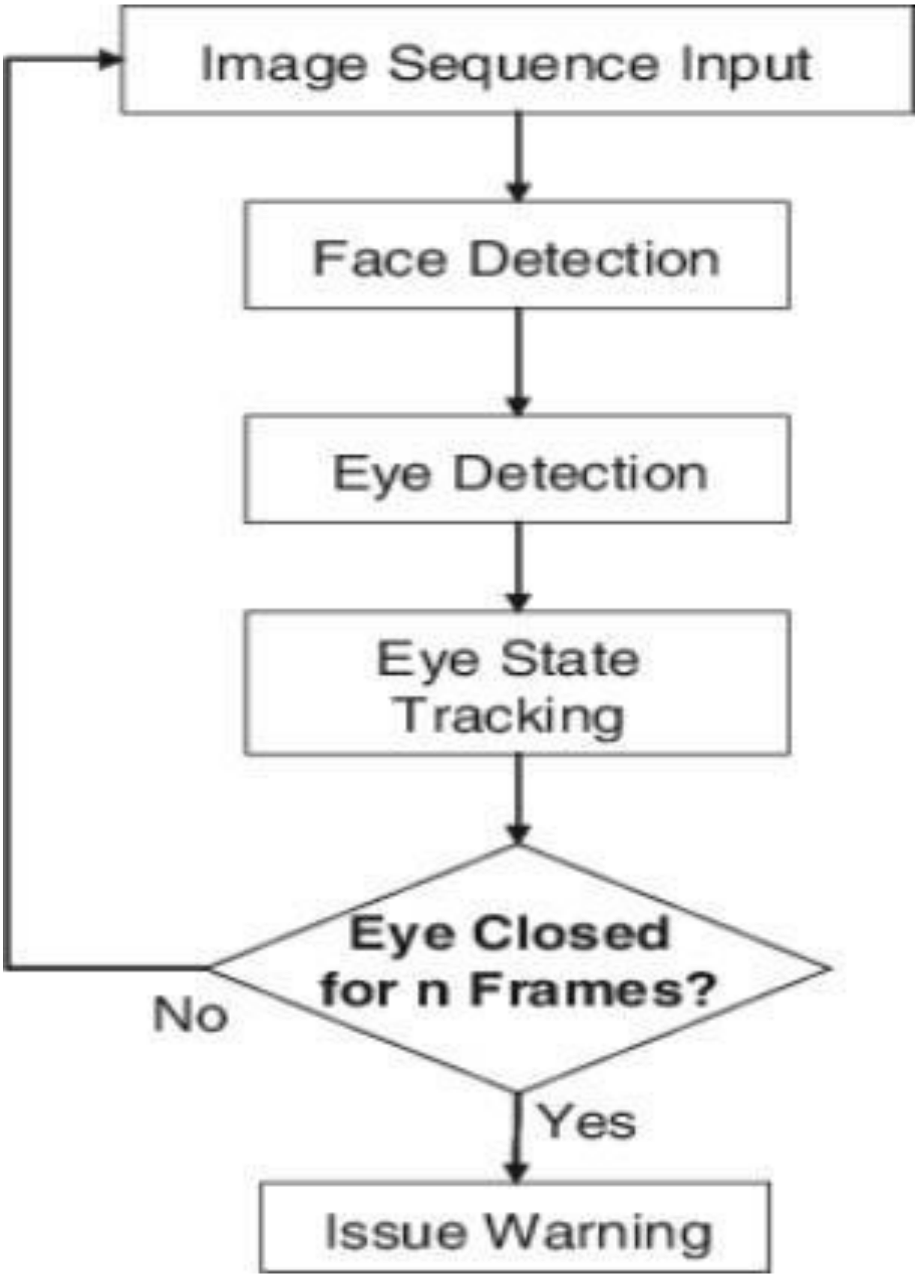


Fig. Methodology

3.2 Computational Analysis :-

- **Drowsiness Detection Design:-** A camera is set up that looks for faces in the input video stream and monitors frames of faces.

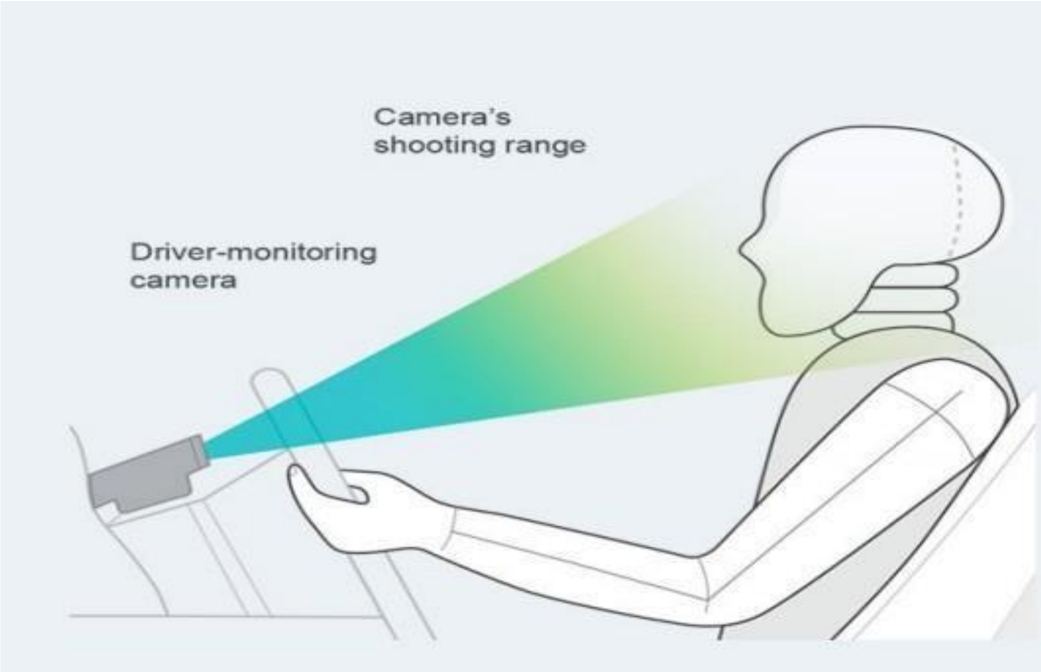


Fig. Practical design of the system

$$\text{EAR} = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

Fig. .Eye Aspect Ratio Calculation Formula

- If the aspect ratio of the eye indicates that they have been closed for greater than a dedicated fixed time, we will sound an alarm system so that the driver wakes up.

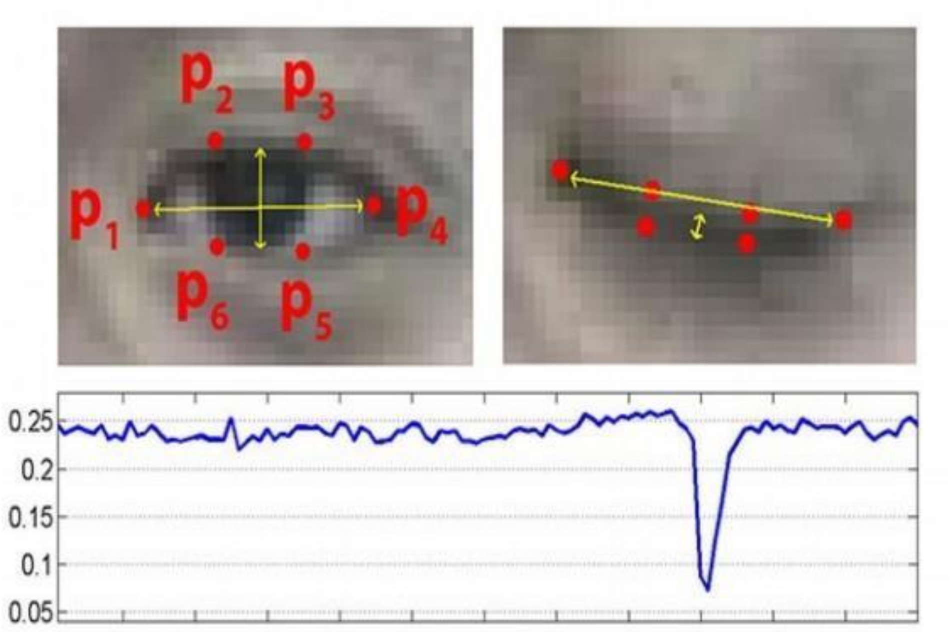


fig:. Eye aspect ratio

- Picture 1 : Eye marks when the eye is open
- Picture 2 : Eye marks when the eye is closed
- Picture 3: Eye Aspect Ratio plotted over time, the downfall in the aspect ratio graph shows a blink of the driver

- We observe the aspect ratio of the eye remains constant for a period of time indicating that the eye was open, then it falls rapidly to zero and then increases again which indicates the person blinked
- We will be observing this eye aspect ratio in our drowsiness detector case to see if the value remains constant or falls to zero but not increases again implying that the driver has closed his eyes for extended period

- dlib library comes with a oriented gradients based face detector histogram a facial landmark predictor comes bundled in the library. Facial landmarks generated by dlib is an indexable list as described in below image

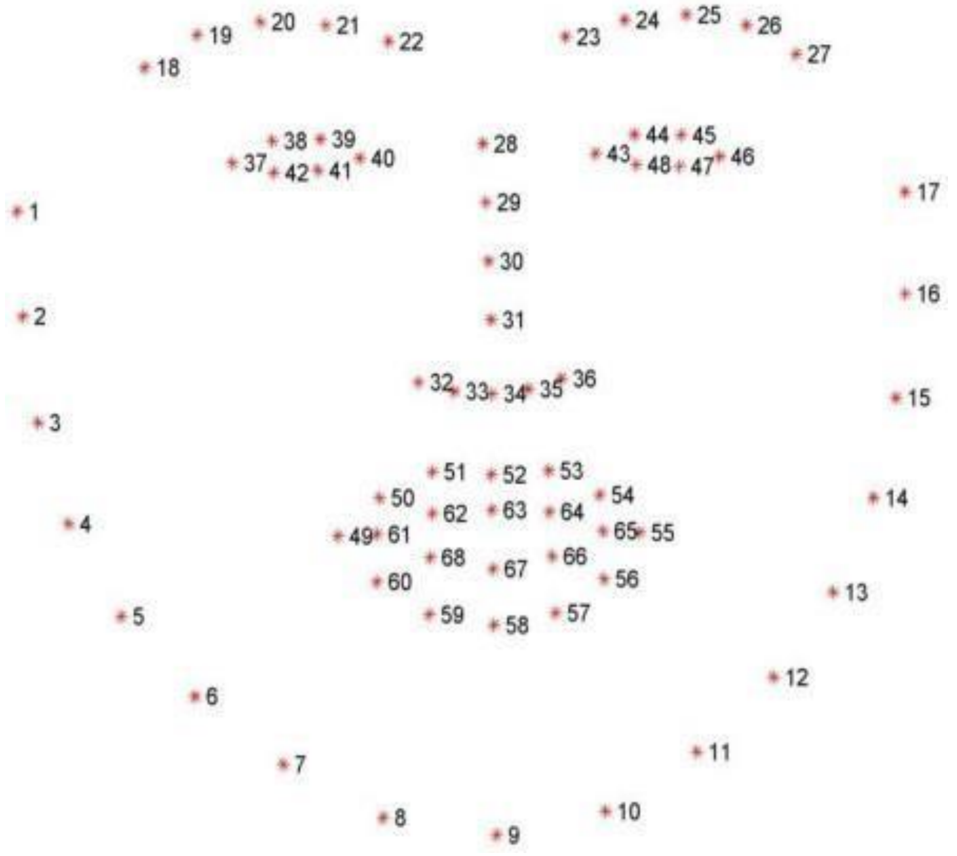
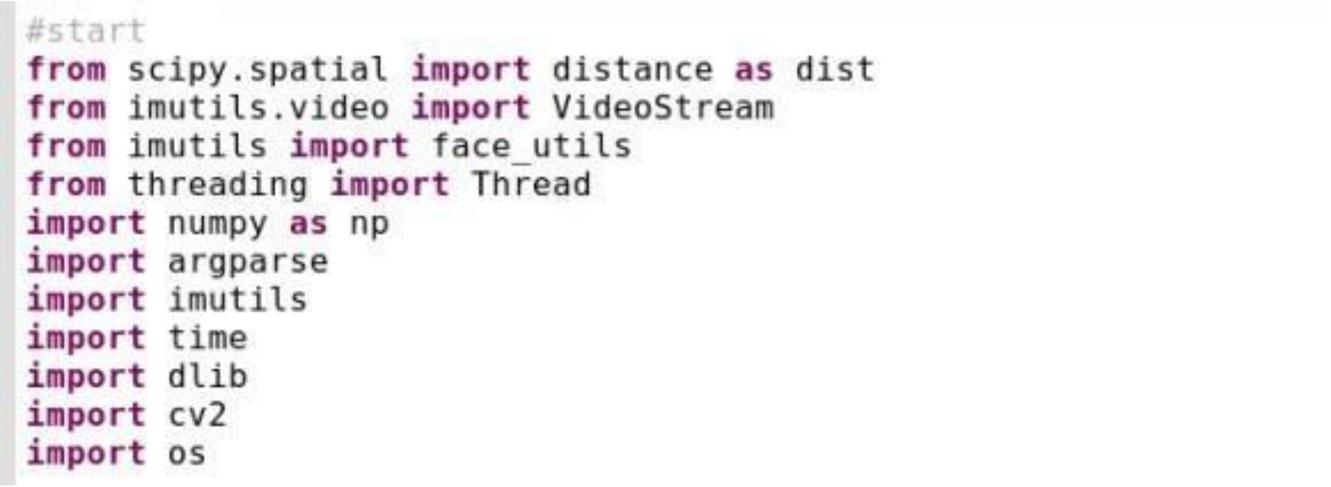


Fig: Facial landmarks set which is detected via dlib

3.3 Starting to build the detector system :-

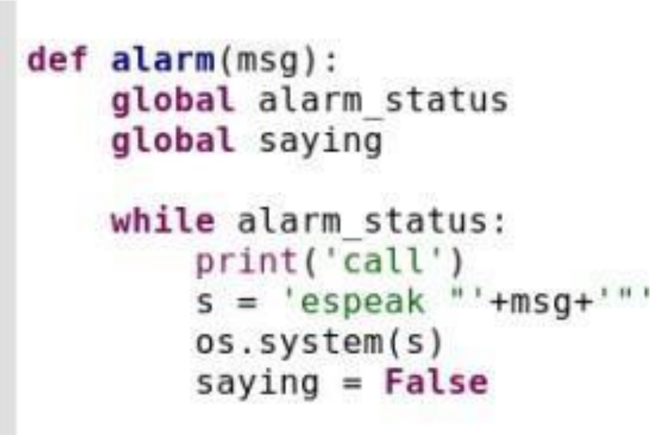
First, we create a new file drowsy_detect.py and write the following script in it : -



```
#start
from scipy.spatial import distance as dist
from imutils.video import VideoStream
from imutils import face_utils
from threading import Thread
import numpy as np
import argparse
import imutils
import time
import dlib
import cv2
import os
```

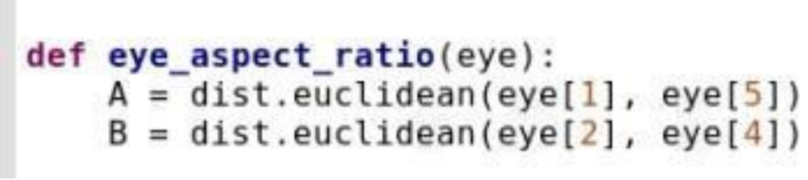
- Now for calculating the eye-aspect ratio we need to compute the Euclidean distance between the facial landmarks points which in turn needs the SciPy package in python.
- Also the package named imutils is needed for image processing and computer vision functions to assist the working with OpenCv.
- The thread class is imported so that we can beep the alarm in a different thread from the main thread so that it is ensured that our script doesn't stop/pause executing while the alarm beeps. In order to play a file of the wav or mp3 format, we need a playsound library.
- Then we created the user defined functions which will be call in main function

alarm (For Warning) :-



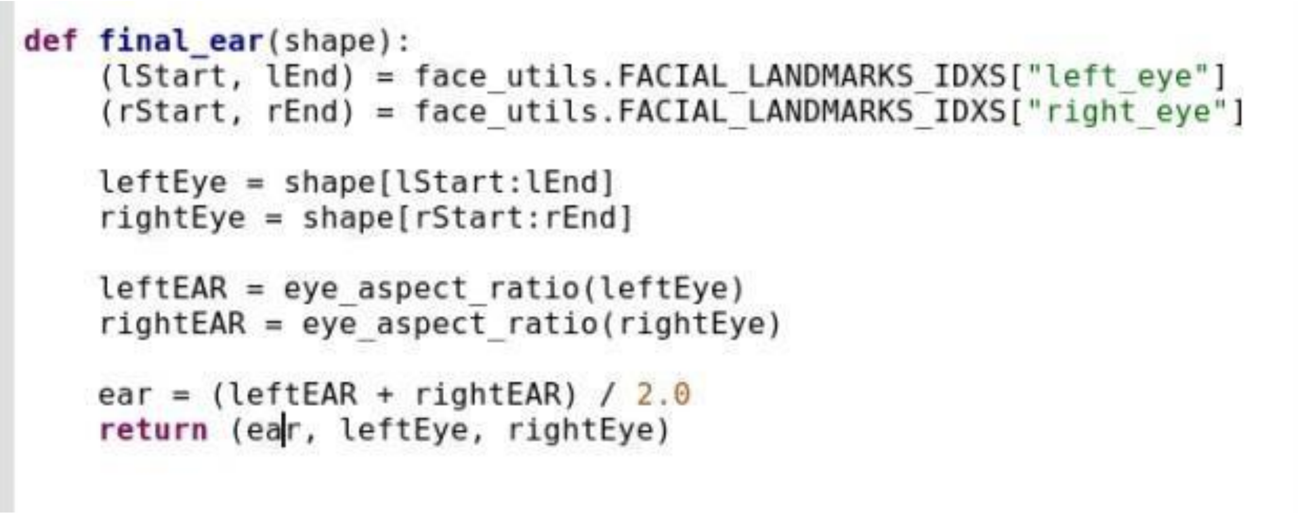
```
def alarm(msg):
    global alarm_status
    global saying

    while alarm_status:
        print('call')
        s = 'espeak "'+msg+'"'
        os.system(s)
        saying = False
```

eye_aspect_ratio(For calculate eye aspect ratio)



```
def eye_aspect_ratio(eye):
    A = dist.euclidean(eye[1], eye[5])
    B = dist.euclidean(eye[2], eye[4])
```

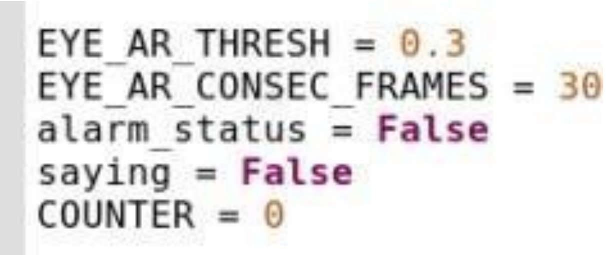
final_ear(To return the final value of EAR)



```
def final_ear(shape):
    (lStart, lEnd) = face_utils.FACIAL_LANDMARKS_IDXS["left_eye"]
    (rStart, rEnd) = face_utils.FACIAL_LANDMARKS_IDXS["right_eye"]

    leftEye = shape[lStart:lEnd]
    rightEye = shape[rStart:rEnd]

    leftEAR = eye_aspect_ratio(leftEye)
    rightEAR = eye_aspect_ratio(rightEye)

    ear = (leftEAR + rightEAR) / 2.0
    return (ear, leftEye, rightEye)
```

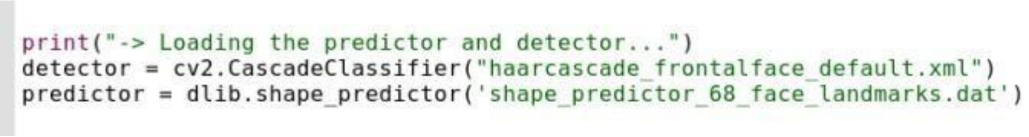
Important variables in the script

- EYE_AR_THRESH: it’s a threshold value for the eye aspect ratio. If ratio becomes lower than this value, the counter starts for the number of frames the eyes remain closed.
- EYE_AR_CONSEC_FRAMES: If the value for the number of frames for which the eyes remain closed exceeds this variable’s value , the alarm is activated.



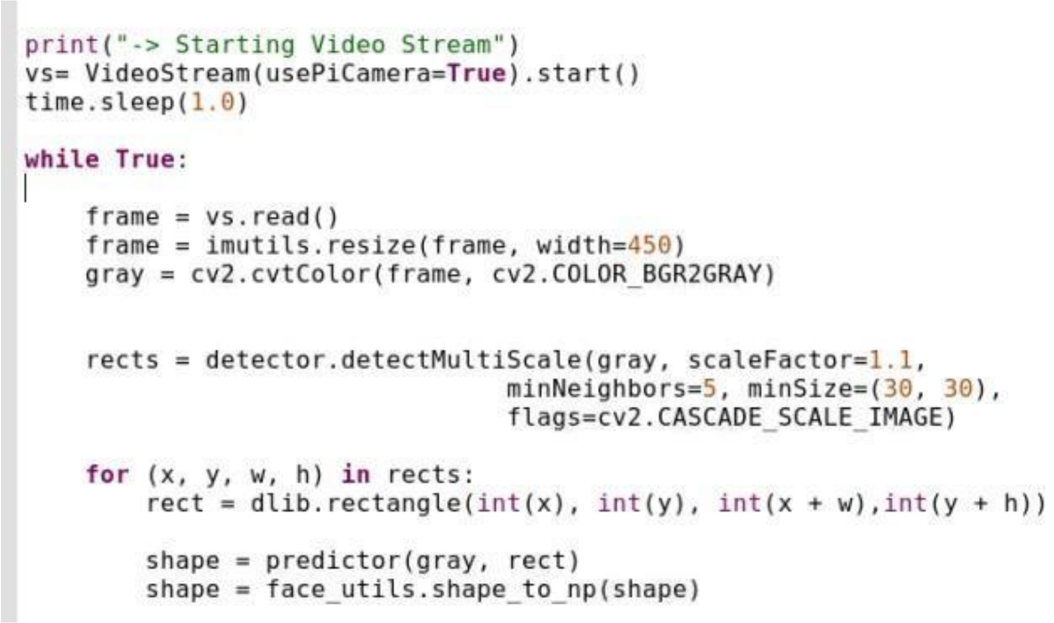
```
EYE_AR_THRESH = 0.3
EYE_AR_CONSEC_FRAMES = 30
alarm_status = False
saying = False
COUNTER = 0
```

calling the detector and predictor file

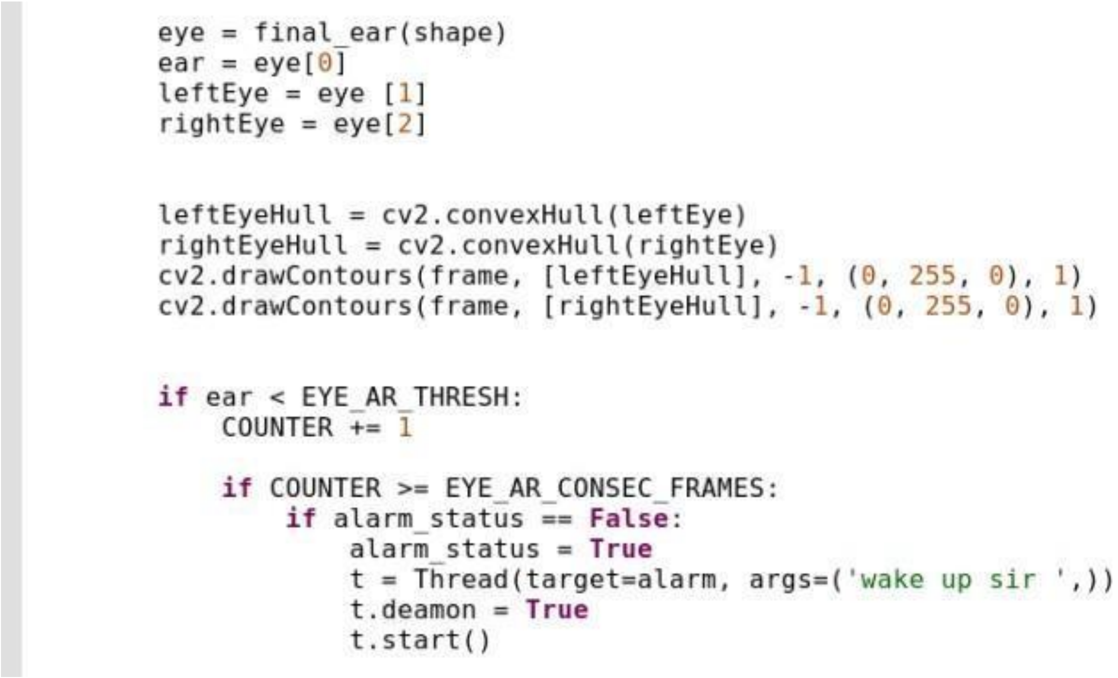


```
print("-> Loading the predictor and detector...")
detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
predictor = dlib.shape_predictor('shape_predictor_68_face_landmarks.dat')
```

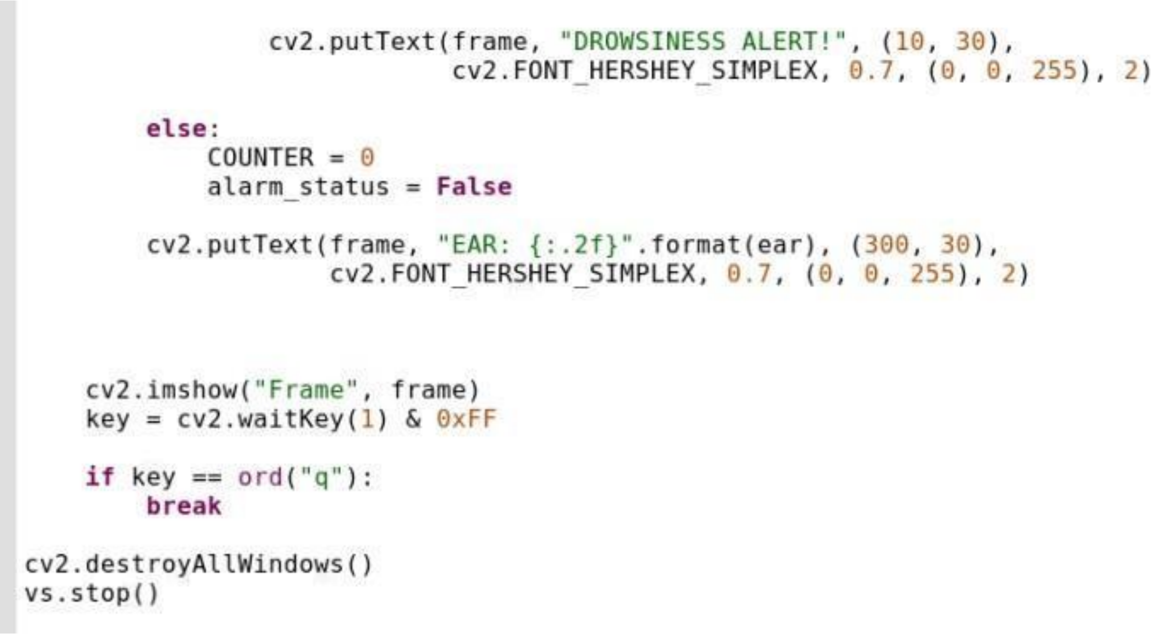
Starting main code



```
print("-> Starting Video Stream")
vs= VideoStream(usePiCamera=True).start()
time.sleep(1.0)

while True:

    frame = vs.read()
    frame = imutils.resize(frame, width=450)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)


    rects = detector.detectMultiScale(gray, scaleFactor=1.1,
                                minNeighbors=5, minSize=(30, 30),
                                flags=cv2.CASCADE_SCALE_IMAGE)

    for (x, y, w, h) in rects:
        rect = dlib.rectangle(int(x), int(y), int(x + w),int(y + h))

        shape = predictor(gray, rect)
        shape = face_utils.shape_to_np(shape)
```



```
        eye = final_ear(shape)
        ear = eye[0]
        leftEye = eye [1]
        rightEye = eye[2]


        leftEyeHull = cv2.convexHull(leftEye)
        rightEyeHull = cv2.convexHull(rightEye)
        cv2.drawContours(frame, [leftEyeHull], -1, (0, 255, 0), 1)
        cv2.drawContours(frame, [rightEyeHull], -1, (0, 255, 0), 1)


        if ear < EYE_AR_THRESH:
            COUNTER += 1

            if COUNTER >= EYE_AR_CONSEC_FRAMES:
                if alarm_status == False:
                    alarm_status = True
                    t = Thread(target=alarm, args=('wake up sir ',))
                    t.deamon = True
                    t.start()
```



```
                cv2.putText(frame, "DROWSINESS ALERT!", (10, 30),
                            cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)

        else:
            COUNTER = 0
            alarm_status = False

        cv2.putText(frame, "EAR: {:.2f}".format(ear), (300, 30),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)


    cv2.imshow("Frame", frame)
    key = cv2.waitKey(1) & 0xFF

    if key == ord("q"):
        break

cv2.destroyAllWindows()
vs.stop()
```

4. TEST

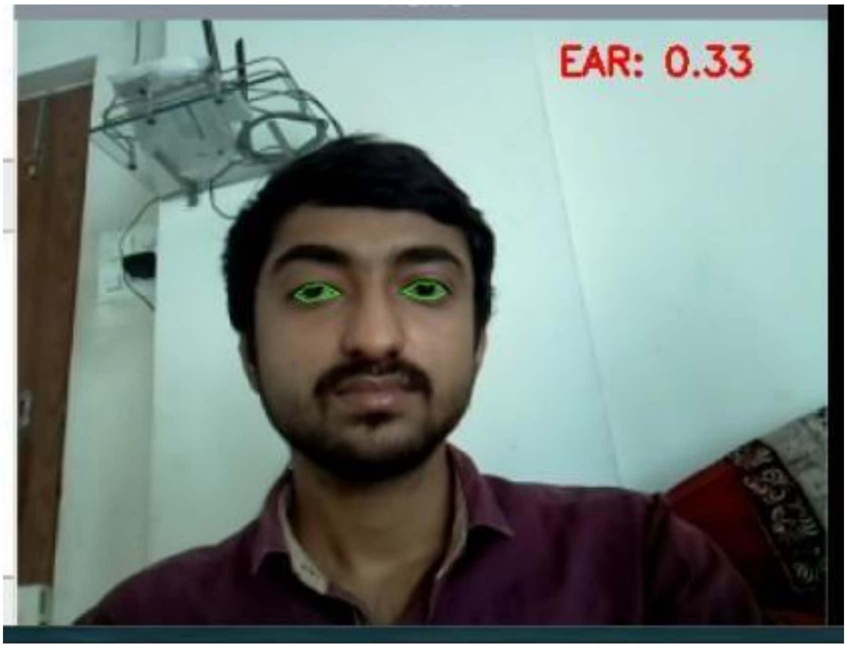
- Test Cases to check the drowsiness Following is the table representing four test cases that are to be encountered while doing this project that concerns the drowsiness of the driver.
- Test Table :-

Test	Eye Detected	Eye Closed	Result
1. Without Glasses	Yes	No	No result
2. Without Glasses	Yes	Yes	Waring
3. With Glasses	Yes	No	No result
4. With Glasses	Yes	Yes	Warning

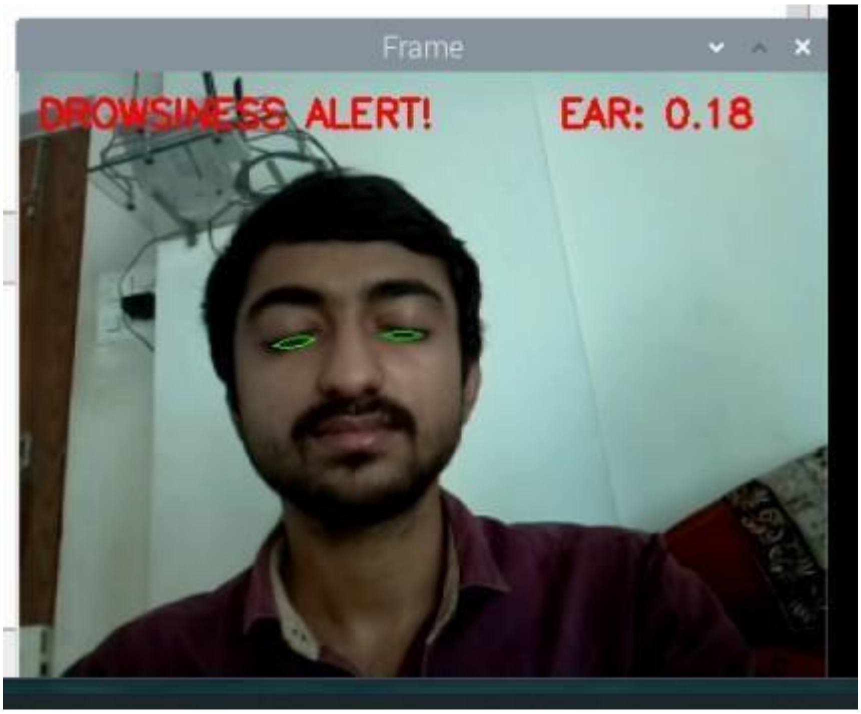
- At the point when the eyes are shut for more than a certain measure of edges then we find that the driver is feeling tired. Henceforth these cases are distinguished is and a caution sounded.

4.1 Test Results :-

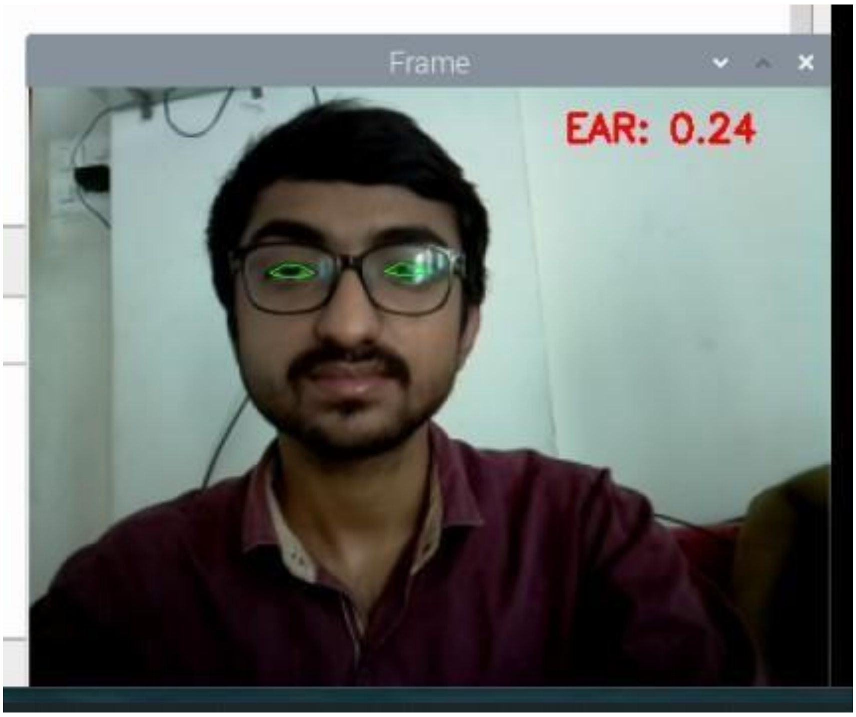
- **Open eye without eyeglasses :-**



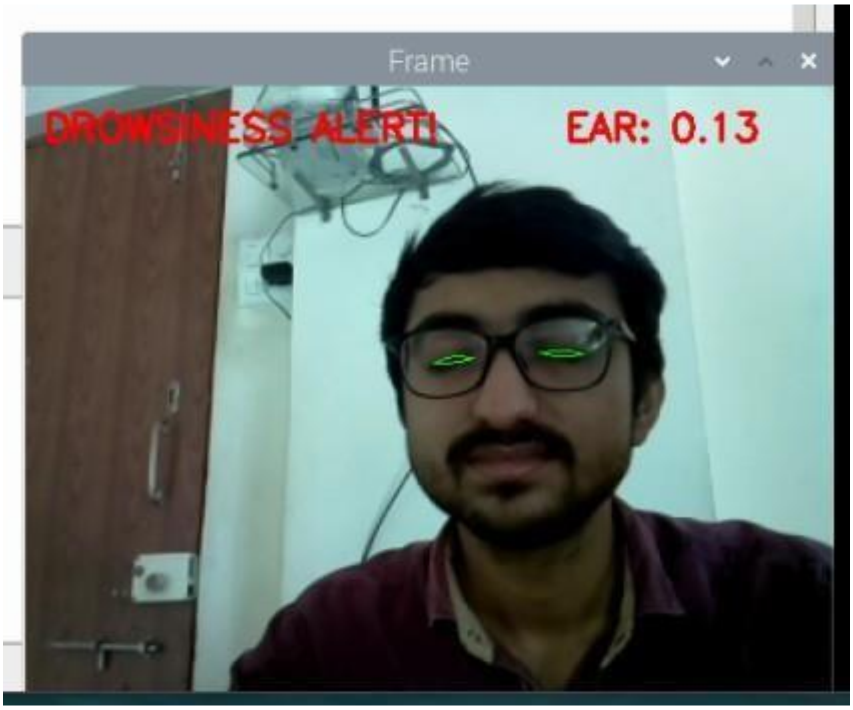
- **Close eye without eyeglasses :-**



Open eye with eyeglasses :-



Close eye with eyeglasses :-



5. CONCLUSION

- Thus we have successfully designed and developed partial implementation of the Driver Drowsiness Detector using Python and OpenCv along with a camera to detect face.
- The system to be developed is to be tested and limitations are identified. The rest of the work will be done according to what is planned already.

6. FUTURE SCOPE

- If there should be an occurrence of an endeavor to robbery, the vehicle's motor doesn't begin or an alarm sounds.
- A picture of the burglar is taken in an attempted theft and sent to the owner of the vehicle who can register a case against the thief of the vehicle.
- It can also be use in public transport for monitoring drivers' condition and sending information to authority.

7. BIBLIOGRAPHY: -

- Raspberry Pi Beginner's Guide 4th Edition
- OPEN/CLOSED EYE ANALYSIS FOR DROWSINESS DETECTION; P.R. TABRIZI AND R. A. ZOROOFI.
- Raspberry Pi Computer Vision Programming: Design and implement computer vision applications with Raspberry Pi, OpenCV, and Python 3, 2nd Edition Kindle Edition by Ashvin Pajanker (Author)

Website Referenced :-

- www.pyimagesearch.com
- learnopencv.com

Reports :-

- www.nsc.org
- ieeexplore.ieee.org
- Sleep-deprived drivers responsible for 40% of road accidents, say transport officials - The Hindu
- 40% of highway accidents occur due to drivers dozing off - The Financial Express