VIT-AP UNIVERSITY

A project report on

Driver drowsiness detection

Batch ID : ECS220474

Under the guidance of

Prof. S Ellison Mathe

Submitted by:

1) MOKARA HEMANT KUMAR 21BCE9109 - SCOPE

2) ALLADA MANASA 21BCE9087 - SCOPE

3) CHOLLANGI PRASANTH 21BCE9339 – SCOPE

4) MARUDI ANJANA DEVI 21BCE9061 – SCOPE

5)PUSULURI JAHNAVI 21BCE9089 - SCOPE

6)BYLAPUDI LAHARI 21BCE9969 - SCOPE

ABSTRACT

Motor vehicle crashes involving drowsy driving are huge in numbers all over the world. Many studies revealed that 10 to 30% of crashes are due to drowsy driving. Fatigue has costly effects on the safety, health, and quality of life. We can detect this drowsiness of driver using various methods, e.g., algorithms based on behavioral gestures, physiological signals and vitals. Also, few of them are vehicle based. Drowsiness of driver was detected based on steering wheel movement and lane change patterns. A pattern is derived based on slow drifting and fast corrective steering movement. We developed a prototype that detects the drowsiness of an automobile driver using artificial intelligence techniques, precisely using opensource tools like OpenCV,dlib on a Raspberry Pi development board. The openCV model is trained on images captured from the video with the help of object detection using Cascade Classifier. In order to have a better accuracy, an Inception v3 architecture is used in pre-training the model with the image dataset. We will give threshold value of eye-aspect ratio based on that raspberry pi will detect the driver drowsiness

ACKNOWLEDGEMENT

In these years of engineering our faculty Dr. Sudha Ellison mathe (Dept of sense) has guided us to achieve a goal and made us capable to complete our Mini Project work successfully. so we would like to take this chance to thank our Project guide, Lab assistants and college with whose support and skill full guidance has made us such as that we would be proud of wherever we go and always be confident in achieving our goals.

Firstly we would like to give our sincere regards to all our Faculty Members of Computer Science and Engineering Department for giving their excellent guidance and who gave us the confidence that we can complete this project successfully. All Faculties of CSE branch have shared their valuable ideas and industrial knowledge about for this project and that's why we are also thanking them for their help. We are thankful to our project guide Dr. Sudha Ellison mathe We are thankful that we got the chance to work under his guidelines and our sincere and heartily regards to him.

We would also like to give vote of thanks to Mr. who has given us permission to carry out this Project work and for providing with the material which initiated our Project work. Finally last but not the least we are very much thank full to our Team Members, Friends for their kind support.

CONTENTS

INDEX **PAGE NO**

1.INTRODUCTION

This driver drowsiness detection project is created to prevent accidents. Drowsiness means sleepiness, so it prevents accidents that are caused by drivers who are feeling drowsy or we can say who fell asleep while driving.

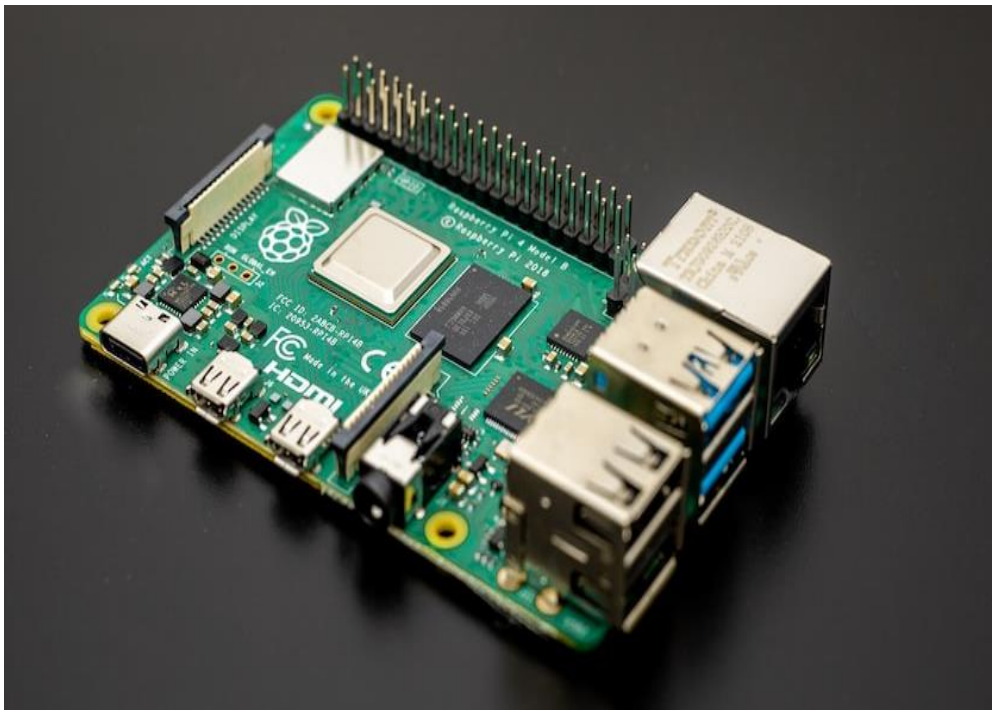
So we are creating a Drowsiness detection system that will detect that the person's eyes are closed or open. And if a person's eyes are closed for a few seconds, the system will alert the person by ringing an alert sound Lack of sleep is the primary reason you're feeling drowsy at the wheel, much less falling asleep while driving. The more sleep debt you accumulate, the more likely you are to get into a road accident caused by drowsy driving. The AAA Foundation for Traffic Safety found that up to 20% of all fatal crashes involve a sleep-deprived driver.

2.BACKGROUND

RASPBERRY PI:

The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python. It's capable of doing everything you'd expect a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games.

Also, the Raspberry Pi has the ability to interact with the outside world, and has been used in a wide array of digital maker projects, from music machines and parent detectors to weather stations and tweeting birdhouses with infra-red cameras. We want to see the Raspberry Pi being used by kids all over the world to learn to program and understand how computers work.



COMMON COMPONENTS OF RASPBERRY PI BOARD:

There are different types of Raspberry pi boards for different purposes. But all the boards have the majority of following components in common.

It comprises a program memory (RAM), processor and graphics chip, CPU, GPU, Ethernet port, GPIO pins, Xbee socket, UART, power source connector. And various interfaces for other external devices. It also requires mass storage, for that we use an SD flash memory card. So that raspberry pi board will boot from this SD card similarly as a PC boots up into windows from its hard disk.

Essential hardware specifications of raspberry pi board mainly include SD card containing Linux OS, US keyboard, monitor, power supply and video cable. Optional hardware specifications include USB mouse, powered USB hub, case, internet connection, the Model A or B: USB Wi Fi adaptor is used and internet connection to Model B is LAN cable.

Memory

The raspberry pi model Aboard is designed with 256MB of SDRAM and model B is designed with 51MB.Raspberry pi is a small size PC compare with other PCs.

The normal PCs RAM memory is available in gigabytes. But in raspberry pi board, the RAM memory is available more than 256MB or 512MB

CPU (Central Processing Unit)

The Central processing unit is the brain of the raspberry pi board and that is responsible for carrying out the instructions of the computer through logical and mathematical operations. The raspberry pi uses ARM11 series processor, which has joined the ranks of the Samsung galaxy phone.

GPU (Graphics Processing Unit)

The GPU is a specialized chip in the raspberry pi board and that is designed to speed up the operation of image calculations. This board designed with a Broadcom video core IV and it supports OpenGL

Ethernet Port

The Ethernet port of the raspberry pi is the main gateway for communicating with additional devices. The raspberry pi Ethernet port is used to plug your home router to access the internet.

GPIO Pins

The general purpose input & output pins are used in the raspberry pi to associate with the other electronic boards. These pins can accept input & output commands based on programming raspberry pi. The raspberry pi affords digital GPIO pins. These pins are used to connect other electronic components. For example, you can connect it to the temperature sensor to transmit digital data.

XBee Socket

The XBee socket is used in raspberry pi board for the wireless communication purpose.

Power Source Connector

The power source cable is a small switch, which is placed on side of the shield. The main purpose of the power source connector is to enable an external power source.

UART

The Universal Asynchronous Receiver/ Transmitter is a serial input & output port. That can be used to transfer the serial data in the form of text and it is useful for converting the debugging code.

Display

The connection options of the raspberry pi board are two types such as HDMI and Composite. Many LCD and HD TV monitors can be attached using an HDMI male cable and with a low-cost adaptor. The versions of HDMI are 1.3 and 1.4 are supported and 1.4 version cable is recommended. The O/Ps of the Raspberry Pi audio and video through HMDI, but does not support HDMI I/p. Older TVs can be connected using composite video. When using a composite video connection, audio is available from the 3.5mm jack socket and can be sent to your TV. To send audio to your TV, you need a cable which adjusts from 3.5mm to double RCA connectors.

NECESSARY COMPONENTS:

Raspberry pi board, speaker, LCD display, webcam, memory card

sandisk

webcam

speaker

3.PROBLEM DEFINITION

1.How to prevent accidents due to drowsiness?

This is particularly applicable to fatigued drivers, who doze off at the wheels between midnight and 5 am on our highways," said B. Shefiq, Joint Regional Transport Officer, Perumbavur.

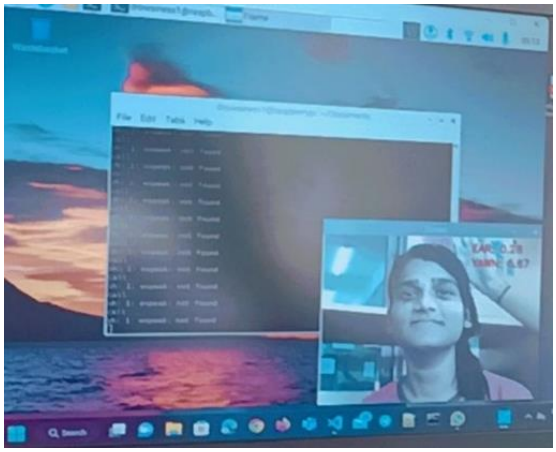
Stating that they usually ascertain the cause of accident owing to sleep deprivation by analysing the tyre markings, Mr. Shefiq said that there will be no signs of the driver applying the brakes near the accident spot as he would have got up only after ramming into another vehicle.

Exhausted drivers who doze off at the wheel are responsible for about 40% of road accidents, says a study by the Central Road Research Institute (CRRI) on the 300-km Agra-Lucknow Expressway.

Each year, drowsy driving accounts for about 100,000 crashes, 71,000 injuries and 1,550 fatalities, according to the National Safety Council (NSC).

4.OBJECTIVE

Many studies revealed that 10 to 30% of crashes are due to drowsy driving. To prevent that we use raspberry pi which will detects the drowsiness of driver and it will wake up him/her



5.METHODOLOGY AND EXPERIMENT

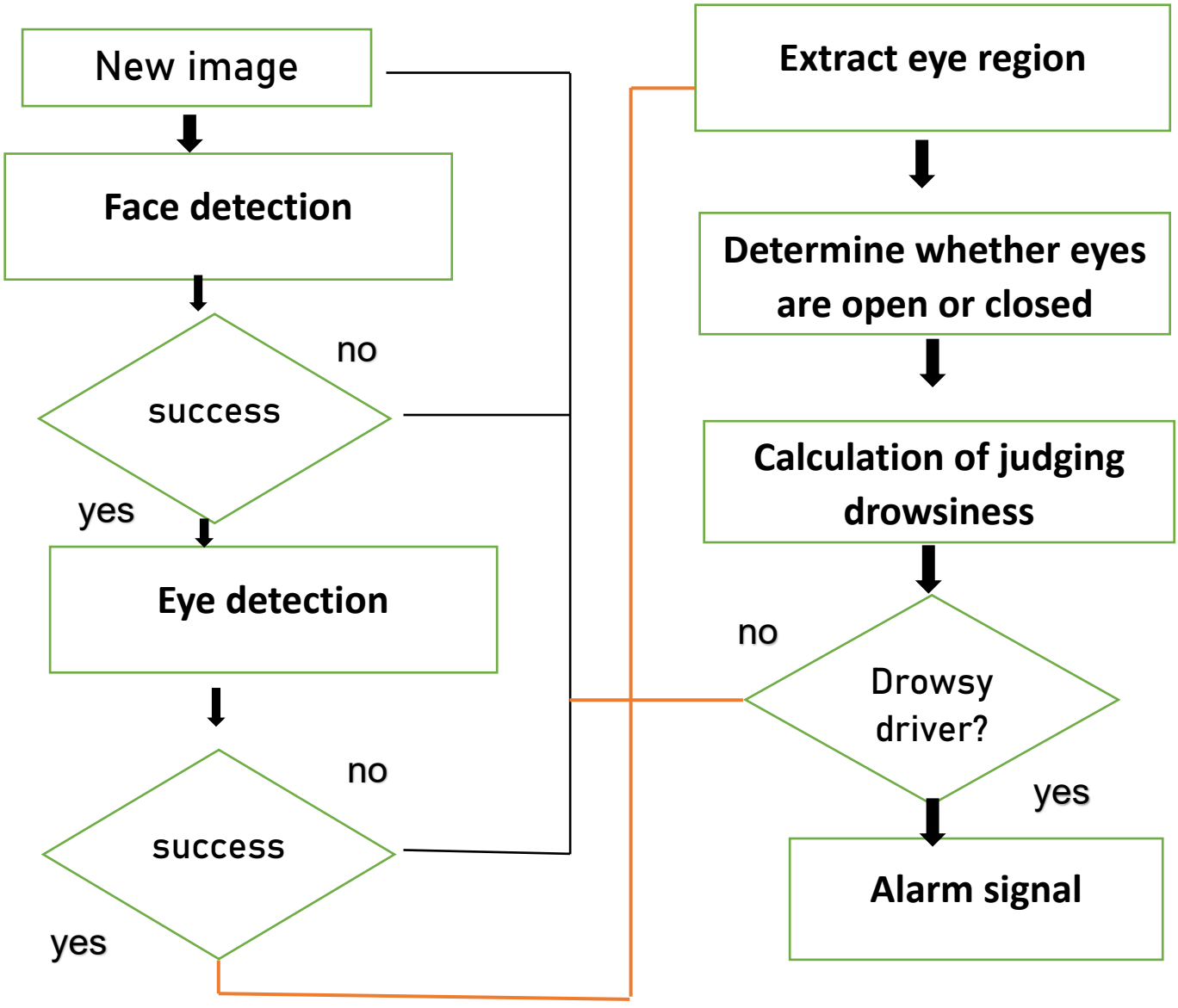
In this work, we focus on behavioral based algorithms using python libraries. We capture the video of driver, convert the video into image frames and deploy our image classification based on ppython modules to detect the level of drowsiness and warn the users using the openCV Lite model on Raspberry Pi 3b board. We used imutils for object detection to convert the video into image frames which consists of left and right eye of the driver and applied eye-aspect ratio formula on the processed images to detect the drowsiness of the driver.

Experimental circuit setup :

Our experiment is based on Raspberry Pi board. We capture the video of the driver using the extended camera on the Raspberry Pi board using OpenCV module of Python and then convert the video into frames and detect the eyes from the frames using Cascade Classifier.

Component	Model	configuration
Board	Raspberry pi 3	Ram – 4gb Memory card-16gb
Camera	Hp webcam	8 Megapixels,720p

Process diagram :

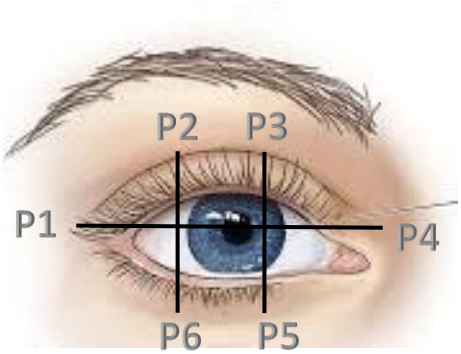


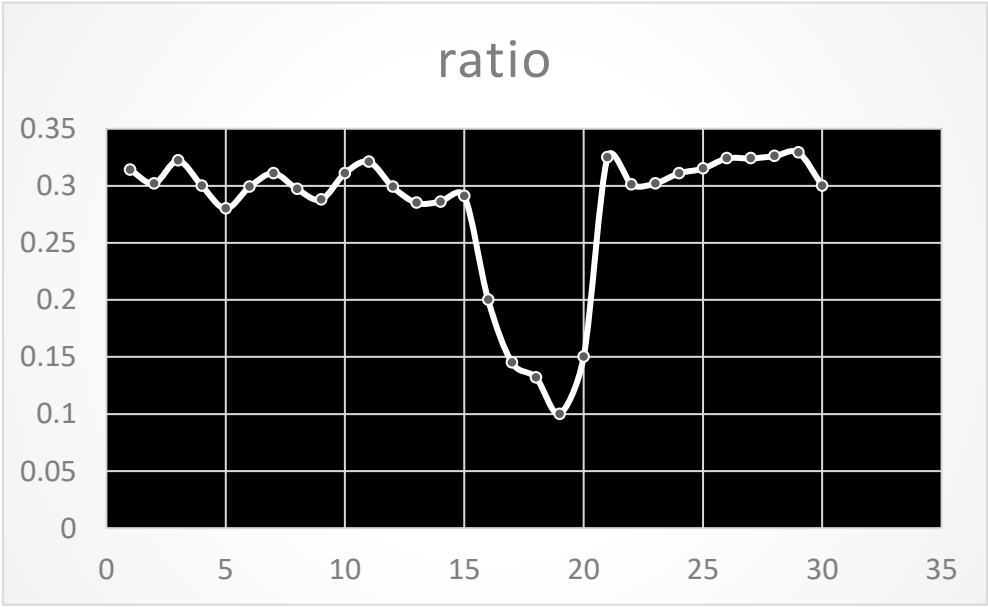
6.OBSERVATION :

1. Here we observe that when the driver starts closing his eyes more than 5secs then speaker will say wake up sir/ma'am..

2. When driver is yawning speaker will say "take a fresh air sir/ma'am
3. This thing will be done when the average eye-aspect ratio of driver is less than threshold value which is 0.28

$$\text{EYE-ASPE CT RATIO} = \frac{|p2-p6| + |p3-p5|}{2\,|p1-p4|}$$





7.RESULT

1. Here when driver sleeping then a text message will be sent to a registered user
2. When driver sleeping then raspberry pi will detect the driver drowsiness and then it process and it will send signal to the speaker

8.CONCLUSION

In this project, we developed a prototype for drowsiness detection for automobile drivers on Raspberry Pi 3. Our prototype captures video of the drivers and feed it to cascade classifier tool. Once the images are identified, they are sent to a pre-trained neural network that accurately detects the level of drowsiness of the driver. imutils train the pattern of drowsiness from the images captured from the video. Note that the training phase is accomplished offline on a desktop. Our prototype on Raspberry Pi 3 is deployable in a wide range of automobile vehicles and can alert the drivers in a non-invasive manner while maintaining a very high level of accuracy. Raspberry Pi 3 gives the edge due to its low cost and energy efficient model

Future Scope :

For now working prototype can detect the drowsiness of the driver by capturing the movements of eyes

Automatically we should stop the modern vehicles when driver was sleeping

We can create an application using cellular networks through satellites for the vehicles owners so that the owner can observe each and every movement of the driver on their mobile phones.

9. REFERENCES

- ✓ https://www.python.org/downloads/release/python-370/
- ✓ https://www.raspberrypi.org/products/
- ✓ https://www.raspberrypi.org/products/camera-module-v2/?resellerType=home
- ✓ https://pyimagesearch.com/2017/05/08/drowsiness-detection-opencv/

10.APPENDIX

- **We imported following modules for this project**

```
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
```

- **This piece of code is to activate the speaker by giving a signal**

```
def alarm(msg):
    global alarm_status
    global alarm_status2
    global saying

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

    if alarm_status2:
        print('call')
        saying = True
```

```
    s = 'espeak "' + msg + '"'
    os.system(s)
    saying = False
```

- **These lines are to calculate the eye-aspect ratio of eyes and mouth(for yawning)**

```
def eye_aspect_ratio(eye):
    A = dist.euclidean(eye[1], eye[5])
    B = dist.euclidean(eye[2], eye[4])
    C = dist.euclidean(eye[0], eye[3])
    ear = (A + B) / (2.0 * C)
    return 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)
def lip_distance(shape):
    top_lip = shape[50:53]
    top_lip = np.concatenate((top_lip, shape[61:64]))
    low_lip = shape[56:59]
    low_lip = np.concatenate((low_lip, shape[65:68]))
    top_mean = np.mean(top_lip, axis=0)
    low_mean = np.mean(low_lip, axis=0)

    distance = abs(top_mean[1] - low_mean[1])
    return distance
```

- **Threshold values for yawning and eye-aspect ratio**

```
EYE_AR_THRESH = 0.3
EYE_AR_CONSEC_FRAMES = 30
YAWN_THRESH = 20
alarm_status = False
alarm_status2 = False
saying = False
COUNTER = 0
```

- **These lines are for face detection and the prediction**

```
print("-> Loading the predictor and detector...")
#detector = dlib.get_frontal_face_detector()
detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")    #Faster but less accurate
predictor = dlib.shape_predictor('shape_predictor_68_face_landmarks.dat')
```

- **These indices will make it simple for us to use an array slice to retrieve the eye regions. We are now ready to start the core of our drowsiness detector:**

```
print("-> Starting Video Stream")
vs = VideoStream(src=args["webcam"]).start()
#vs= VideoStream(usePiCamera=True).start()       //For Raspberry Pi
time.sleep(1.0)
while True:
   frame = vs.read()
   frame = imutils.resize(frame, width=450)
   gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
   #rects = detector(gray, 0)
   rects = detector.detectMultiScale(gray, scaleFactor=1.1,
            minNeighbors=5, minSize=(30, 30),
```

```
                flags=cv2.CASCADE_SCALE_IMAGE)
```

- **The face detector from dlib is then used to discover and pinpoint the face(s) in the picture. The next step is to apply facial landmark detection to localize each of the important regions of the face:**

```
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]
  distance = lip_distance(shape)
```

- **Using the cv2.drawContours method, we can then see each of the eye areas on our frame. This is useful when we are attempting to debug our script and want to make sure that the eyes are being appropriately recognised and localised:**

```
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)
lip = shape[48:60]
cv2.drawContours(frame, [lip], -1, (0, 255, 0), 1)
```

- **Finally, we are now ready to check to see if the person in our video stream is starting to show symptoms of drowsiness:**

```
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
    if (distance > YAWN_THRESH):
        cv2.putText(frame, "Yawn Alert", (10, 30),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
        if alarm_status2 == False and saying == False:
            alarm_status2 = True
            t = Thread(target=alarm, args=('take some fresh air sir',))
            t.deamon = True
            t.start()
    else:
      alarm_status2 = False
```

- **The final code block in our drowsiness detector handles displaying the output Frame to our screen:**

```
      cv2.putText(frame, "EAR: {:.2f}".format(ear), (300, 30),
                  cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
      cv2.putText(frame, "YAWN: {:.2f}".format(distance), (300, 60),
                  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()
```