DRIVER DROWSINESS DETECTION SYSTEM FOR ACCIDENT PREVENTION

Submitted By

Team – 06

Menta Sai Akshay (CB.EN.U4AIE20040)

Rachure Charith Sai (CB.EN.U4AIE20055)

Sajja Bala Karthikeya (CB.EN.U4AIE20065)

Penaka Vishnu Reddy (CB.EN.U4AIE20048)

Under the supervision of

B. Ganga Gowri (CEN)

For the Completion of

19AIE211 - Introduction to Computer Networks

CSE - AI

18th July 2022

ACKNOWLEDGEMENT

It would be my immense pleasure to take this as a chance to represent my appreciation and earnestness because of our regarded **B. Ganga Gowri (CEN)** for the direction, knowledge, and bolster that he has given all throughout this work. My present work would never have been achievable without his guidance input and tutoring.

I would also like to thank every one of my friends, faculty team, and all staff of the prestigious Department of **Centre for Excellence in Computational Engineering and Networking**, Amrita Vishwa Vidyapeetham, Coimbatore for their amazing help all through our course of learning at this organization.

CONTENTS: Page

ABSTRACT

In recent years, driver fatigue is one of the world's major causes of vehicle accidents. A direct way of measuring driver fatigue is by measuring the driver’s state, i.e., drowsiness. So, it is very important to detect the drowsiness of the driver to save life and property. This project is aimed at developing a prototype of the drowsiness detection system. This system is a real-time system that continuously captures images, measures the state of the eye according to the specified algorithm, and gives a warning if required.

For detection of drowsiness the per closure value of eye is considered. So when the closure of eye exceeds a certain amount then the driver is identified to be sleepy. For implementing this system several OpenCV libraries are used including Haar-cascade. The entire system is implemented using Raspberry-Pi.

Chapter 1

1.1 Introduction

The attention level of driver degrades because of less sleep, long continuous driving or any other medical condition like brain disorders etc. Several surveys on road accidents says that around 30 percent of accidents are caused by fatigue of the driver. When driver drives for more than normal period for human then excessive fatigue is caused and also results in tiredness which drives the driver to sleepy condition or loss of consciousness.

Drowsiness is a complex phenomenon that states that there is a decrease in alerts and conscious levels of the driver. Though there is no direct measure to detect drowsiness several indirect methods can be used for this purpose.

In chapter 1, in the initial sections, different types of methods for measuring the drowsiness of the driver are mentioned which include Vehicle based measures, Physiological measures, Behavioral measures. Using those methods an intelligence system can be developed which would alert the driver in case drowsy condition and prevent accidents. Advantages and dis advantages corresponding to each and every system is explained. Depending on advantages and disadvantages the most suitable method is chosen and proposed. Then the approach for entire system development is explained using a flow chart which includes capturing the image in real time continuously, then dividing it into frames. Then each frame is analyzed to find face first. If a face is detected then then next task is to locate the eyes. After the positive result of detecting the eye the amount of closure of eye is determined and compared with the reference values for the drowsy state eye. If a drowsy condition is found out then the driver is alarmed else repeatedly the loop of finding the face and detecting the drowsy condition is carried out.

In the latter sections object detection, face detection and eye detection and eye detection are explained in a detailed manner. Because the face is a type of object hence a few studies on object detection is done. In face detection and eye detection, different approaches for both are proposed and explained.

Chapter 2

2.1 Drowsiness

Drowsiness is defined as a decreased level of awareness portrayed by sleepiness and trouble in staying alarmed but the person awakes with simple excitement by stimuli. It might be caused by an absence of rest, medicine, substance misuse, or a cerebral issue. It is mostly the result of fatigue which can be both mental and physical. Physical fatigue, or muscle weariness, is the temporary physical failure of a muscle to perform ideally. Mental fatigue is a temporary failure to keep up with ideal psychological execution. The onset of mental exhaustion amid any intellectual action is progressive and relies on an individual's psychological capacity, furthermore upon different elements, for example, lack of sleep and general well-being. Mental exhaustion has additionally appeared to diminish physical performance. It can show as sleepiness, dormancy, or coordinated consideration weakness.

In the past years according to available data driver sleepiness has gotten to be one of the real reasons for street mishaps prompting demise and extreme physical injuries and loss of economy. A driver who falls asleep is on the edge of losing control over the vehicle prompting a crash with other vehicles or stationary bodies. Keeping in mind to stop or reduce the number of accidents to a great extent the condition of sleepiness of the driver should be observed continuously.

2.2 Measures for detection of Drowsiness

The study states that the reason for a mishap can be categorized as one of the accompanying primary classes: (1) human, (2) vehicular, and (3) surrounding factors. The driver's error represented 91% of the accidents. The other two classes of causative elements were referred to as 4% for the type of vehicle used and 5% for surrounding factors.

Several measures are available for the measurement of drowsiness which includes the following:

1. **Vehicle-based measures**
2. **Physiological measures**
3. **Behavioral measures**

1. **Vehicle-based measures**
   Vehicle-based measures survey path position, which monitors the vehicle's position as it identifies with path markings, to determine driver weakness, and accumulate steering wheel movement information to characterize the fatigue from low level to high level. In many research projects, researchers have used this method to detect fatigue, highlighting the continuous nature of this non-intrusive and cost-effective monitoring technique.

   This is done by:

   1. Sudden deviation of the vehicle from lane position.
   2. Sudden movement of steering wheels.
   3. Pressure on acceleration paddles.

2. **Physiological measures**
   Physiological measures are the objective measures of the physical changes that occur in our bodies because of fatigue. These physiological changes can be simply measured by their respective instruments as follows:

   ECG (electro cardiogram)
   EMG (electromyogram)
   EOG (electro oscillogram)
   EEG (electroencephalogram)

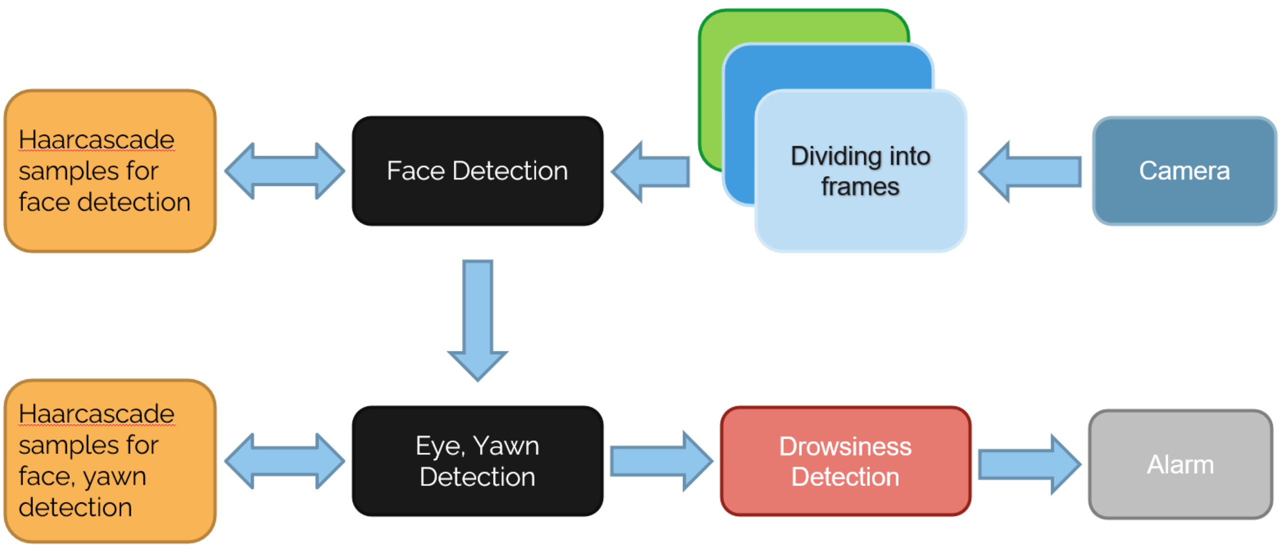
3. **Behavioral measures**
   Certain behavioral changes take place during drowsing like
   1. Yawning
   2. Amount of eye closure
   3. Eye blinking
   4. Head position

2.3 Proposed Method

Among all these four strategies, the most precise technique depends on human physiological measures. This procedure is executed in two ways: measuring changes in physiological signs, for example, brain waves, heart rate, and eye flickering; and measuring physical changes, for example, sagging posture, and open/shut conditions of the eyes. In spite of the fact that this procedure is most precise, it is not reasonable, since detecting electrodes would need to be put straight onto the driver's body, and thus be irritating and diverting to the driver. Also, long time driving would bring about sweat on the sensors, reducing their capacity to screen precisely.

Hence this approach will mostly focus on the amount of eye closure also called (PERCLOS) percentage of closure as it provides the most accurate information on drowsiness. It is also non-intrusive in nature, hence does not affect the state of the driver, and also the driver feels totally comfortable with this system. Environmental factors like road conditions do not affect this system. The case of micro nap is also detected according to the given threshold value. The development of this system includes face identification and tracking, detection and location of the human eye, human eye tracking, eye state detection, and driver fatigue testing. The key parts of the detection framework fused the detection and location of human eyes and driver fatigue testing. The improved technique for measuring the PERCLOS estimation of the driver was to compute the proportion of the eyes being open and shut with the aggregate number of frames for a given period.

2.4 Drowsiness detection approach



2.5 Algorithm Stages

Image Capture:

Utilizing a web camera introduced inside the automobile we can get the picture of the driver. Despite the fact that the camera creates a video clip, we have to apply the developed algorithm on each edge of the video stream.

Dividing into Frames:

We are dealing with real time situation where video is recorded and has to be processed. But the processing or application of algorithm can be done only on an image. Hence the captured video has to be divided into frames for analyzing.

Face Detection:

In this stage we detect the region containing the face of the driver. A specified algorithm is for detection of face in every frame.

By face detection we means that locating the face in a frame or in other words finding location of facial characters through a type of technology with the use of computer.

The frame may be any random frame. Only facial related structures or features are detected and all others types of objects like buildings, tree, bodies are ignored.

Eye Detection:

After successful detection of face eye needs to be detected for further processing. In our method eye is the decision parameter for finding the state of driver.

Though detection of eye may be easier to locate, but it’s really quite complicated. At this point it performs the detection of eye in the required particular region with the use of detection of several features.

Generally Eigen approach is used for this process. It is a time taking process. When eye detection is done then the result is matched with the reference or threshold value for deciding the state of the driver.

Yawn Detection:

To detect the yawn, we need to find out the distance between the user's upper lip and lower lip. So, when a person is talking this distance will be within a limit, but when the person takes yawns, the distance will be much higher than the limit or threshold.

Now to find out the distance between two lips, we need to find out the landmarks of the lips and again we will use the DLIB's facial landmark model here.

State of eye:

In this stage, we find the actual state of the eye that if it is closed or open or semi closed or open. The identification of eyes status is most important requirement. It is achieved by an algorithm which will be clarified in the later parts. We channelize a warning message if we obtain that the eyes are in open state or semi open state up to a particular threshold value. If the system detects that the eyes are open then the steps are repeated again and again until it finds a closed eye.

Chapter 3

3.1 Introduction

This chapter deals initially with object detection. Because face is also a type of object, hence how the detection of object is done using OpenCV is described. Next it deals with face detection techniques based on object detection. Though several class of objects can be identified using object detection technique but for our purpose only face detection will be used. Next part of this chapter focus on eye detection technique as it is the most important stage drowsiness detection and also the next step of measurement of state of eye strongly depends on this.

3.2 Object Detection

Object detection is commonly defined as method for discovering and identifying the existence of objects of a certain class. Also it can be considered as a method in image processing to find out an object from images. There are several ways to classify and find objects in a frame. Out of that one way can be based on color identification. But it is not an efficient method to detect the object as several different size object of same color may be present. Hence a more efficient way is Haar-like features, developed by Viola and Jones on the basis of the proposal by Papageorgiou et. al in 1998. Haar-like features are digital image features used in object detection. Or we can say that these are rectangle shaped dark and light areas having similar kind of features like our face. The cascade classifier comprises of a number of stages, where each stage consists of many weak features. The system detects objects by moving a window over the entire image and by forming a strong classifier. The output of each stage is labeled as either positive or negative– positive meaning that an object was found and negative means that the specified object was not found in the image.

3.3 Face detection

We know that face is also a type of object. So we can consider detection of face as a particular case of object detection. In this type of object type of class detection, we try to know where the objects in the interest image are located and what is their size which may belongs to a particular class. The work of algorithm that is made for face detection is mostly concentrated on finding the front side of the face. But the algorithm that are developed recently focus on more general cases. For our case it may be face in the tilted position or any other portion of the faces and also it finds the possibility of multiple faces. Which means the rotation axis with respect to the present observer from the reference of face in a particular.

Or even if there is vertical rotation plane then also it is able to solve the purpose. In new type of algorithm, it is considered that the picture or video is a variable which means that different condition in them like hue contrast may change its variance. The amount of light may also affect. Also, the position of the input may vary the output. Many calculations actualize the face-detection assignment as a two-way pattern-differentiation task. It means the contextual features present in the interest image is repeatedly change into features and this results in preparing the respective classifier on the reference faces which decides if the specified area is a face or any other objects. If we obtain a positive response for the detecting a face then the process goes for next stage continuation otherwise the algorithm is designed in such manner to go for capturing of image till any hint of face is found. The main algorithm used for this process is Viola Jones algorithm. For getting particular output the utilization of cascade part of open cv is made. Cascade file of OpenCV contains 24 stages and has got 2913 weak classifiers. Its window starts with size of 24 x 24 pixels. Set up for the starting scale has to be made 1.0 and the step size of each scale was set to 1.1 and the position step size Δ was set to 1.0. The total number of scales used is 32 resulting in a total of more than 1.8 million possible detection window which is huge. Training of cascade was done by OpenCV hence it is easy to use.

3.4 Eye detection

Poor contrast of eyes generally creates a lot of problems in its detection. After successful detection of face eye needs to be detected for further processing. In our method eye is the decision parameter for finding the state of driver. Though detection of eye does not look complex but the actual process is quite hectic. In this case it performs the detection of eye in the specified region with the use of feature detection. Generally, Eigen approach is used for this process. It is a time taking process. When eye detection is done then the result is matched with the reference or threshold value for deciding the state of the driver. Eye detection is divided into two categories: eye contour detection and eye position detection. Basically, eyes are detected based on the assumption that they are darker than other part of the face. Hence Haar Features of similar type can be moved throughout the upper part of the face to match with the feature of eye leading to location of eye. We consider as potential eye areas, the non-skin locales inside face district. Clearly, eyes ought to be inside a face area and eyes are not distinguished as skin by the skin identifier. In this way, we need to discover eye-simple sets among a decreased number of potential eye regions. In recent years several eye detection methods have been developed. Deformable template is one of the popular methods in identifying the human eye. In this method, a model of eye is designed first and then eye position is obtained by recursive method. But this method strongly depends on initial position of the eye which should be near the actual position of eye. In the template matching aspect, the proposed algorithm is based on eigenfeatures and neural networks for the extraction of eyes using rectangular fitting from Gray-level face images. This method does not need a large set of training images in its advantage and does by eigenfeatures and sliding window. But this algorithm fails if the user uses glasses or having beard. We know that using Haar features in AdaBoost results in increasing computational efficiency and accuracy than other methods for face detection. But Haar feature has a limitation i.e., discriminant capability. Although the Haar features vary with different patterns, sizes and positions, they can only represent the regular rectangular shapes. But for our case of eye detection eye and iris is of round shape. Hence eyes can be represented by learning discriminate features to characterize eye patterns. So an approach towards probabilistic classifier to separate eyes and non-eyes are much better option for better accuracy and for robustness.

Chapter 4

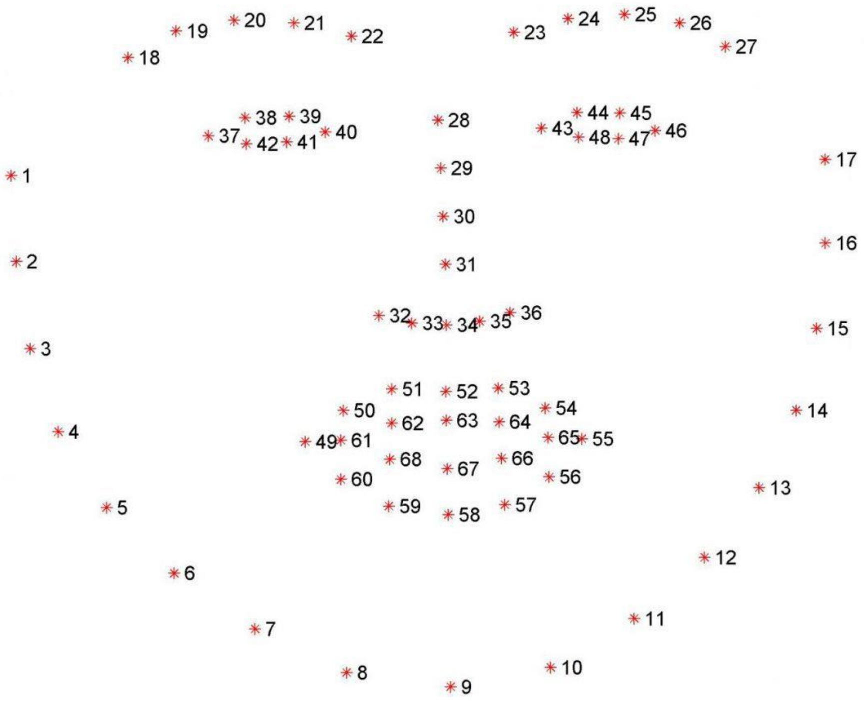
4.1 Introduction

Chapter 4 deals with the theoretical and mathematical explanation of the various approach made for face and eye detection.

4.2 Approach

To start, we will apply OpenCV's Haar cascades to detect the face in an image, which boils down to finding the bounding box (x, y)-coordinates of the face in the frame.

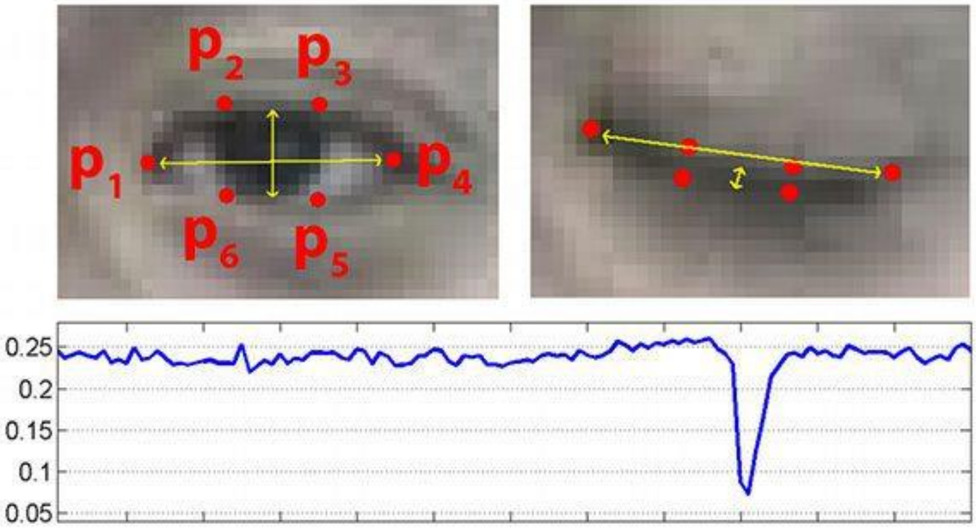
Given the bounding box of the face we can apply dlib's facial landmark predictor to obtain 68 salient points used to localize the eyes, eyebrows, nose, mouth, and jawline:



Visualizing the 68 facial landmark coordinates from the iBUG 300-W dataset.

As we discussed, dlib's 68 facial landmarks are indexable which enables us to extract the various facial structures using simple Python array slices.

Given the facial landmarks associated with an eye, we can apply the Eye Aspect Ratio (EAR) algorithm which was introduced by Soukupová and Čech in their 2017 paper,

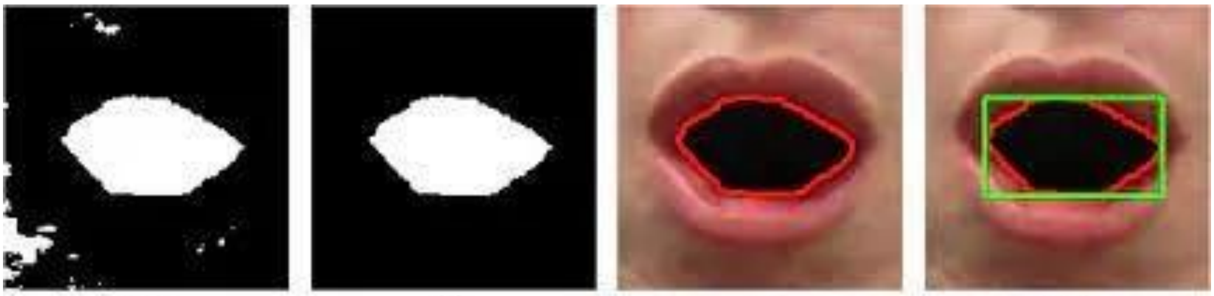


On the *top-left*, we have an eye that is fully open and the eye's facial landmarks plotted. Then on the *top-right*, we have an eye that is closed. The *bottom* then plots the eye aspect ratio over time. As we can see, the eye aspect ratio is constant (indicating that the eye is open), then rapidly drops to close to zero, then increases again, indicating a blink has taken place.

Finally, after obtaining the points, we can find the EAR, using the formula.

$$EAR = \frac{(|P_2 - P_6| + |P_1 - P_5|)}{(2 * |P_1 - P_4|)}$$

Next, we will check if this EAR value is within the threshold or not, and based on that the system will alert the user.



(a) (b) (c) (d)

To detect the yawn, we need to find out the distance between the user's upper lip and lower lip. So, when a person is talking this distance will be within a limit, but when the person takes yawns, the distance will be much higher than the limit or threshold.

Now to find out the distance between two lips, we need to find out the landmarks of the lips and again we will use the DLIB's facial landmark model here.

Then we will simply calculate the distance between the midpoint of the upper lip to the midpoint of the lower lip. And if this distance is more than the threshold, the system will give a yawn alert to the user.

Chapter 5

5.1 Introduction

Chapter 5 includes the implementation of the drowsiness detection system with the hardware. The hard ware used is Raspberry Pi. So little description of the used hardware with its features and its installation and setup procedure are also described. Mid portion of the chapter described how the entire process of drowsiness detection occurs in low level. For conducting this libraries of OpenCv is used. Different .xml files of OpenCv is operated on the input and provide the required result. The .xml files written for drowsiness detection includes face and eye detection which basically done by algorithm developed by Viola-Jones. Those algorithm includes Haar features, Formation of integral Image, Adaboost and Cascading. Theoretical part of all those features are described briefly.

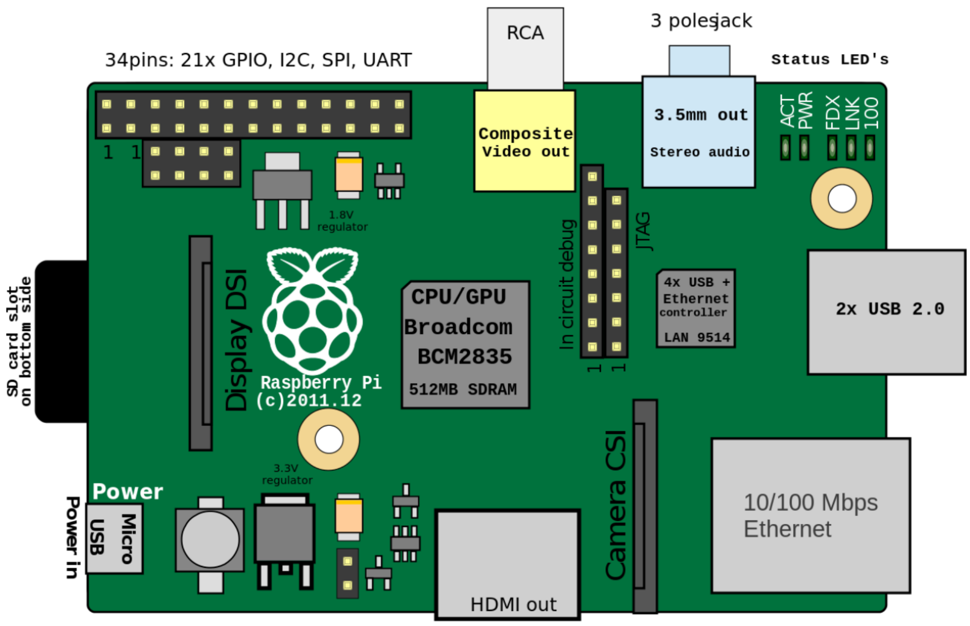
5.2 Implementation

5.2.1 About Raspberry Pi

It is a low cost, credit-card sized computer which is used for implementing small projects. A monitor or TV has to be connected with it externally to visualize its operating system and operate it. We can use a key board and a mouse to provide input to it. An external memory has to be used to load its operating system. We can program it with several languages like C++, Python etc.

Its components includes the following:

1. 700 MHz processor
2. 512 MB RAM
3. USB ports for external devices
4. Micro SD card slots
5. Ethernet port
6. HDMI port
7. 40 GPIO pins
8. Camera interface
9. Display interface
10. Power supply port



We have implemented Face detection with the help of

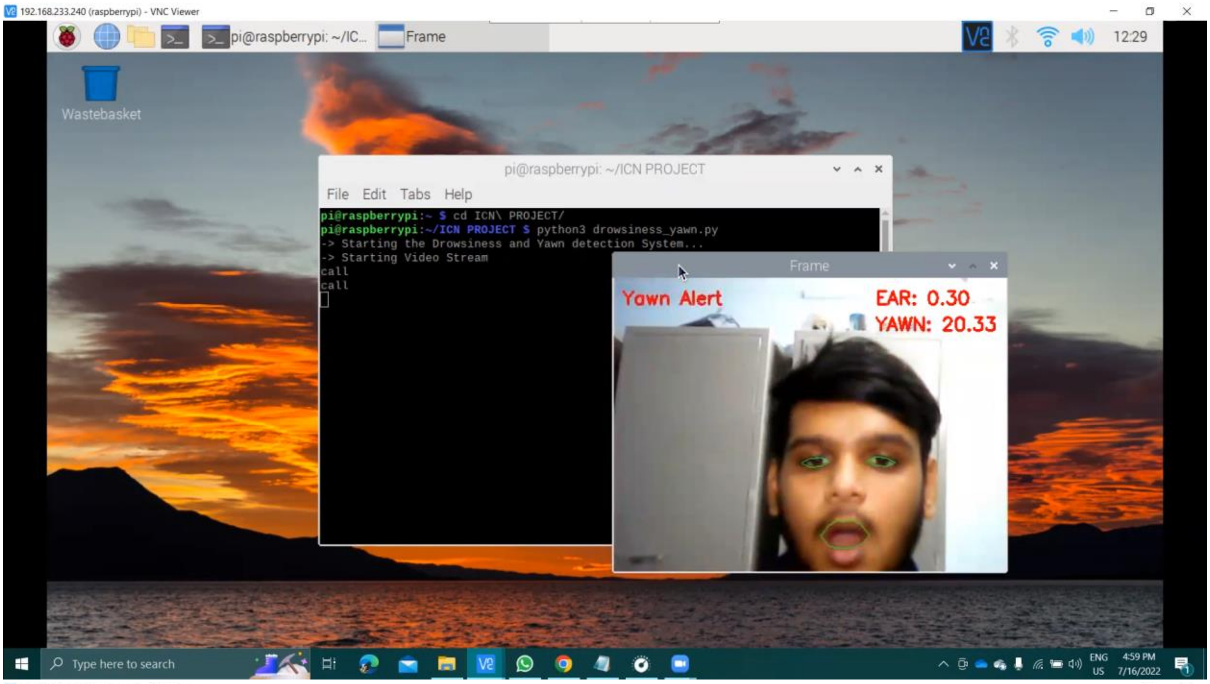
1. Raspberry pi
2. Web Cam
3. Raspbian operating system
4. Python IDLE
5. OpenCV (Open source Computer Vision) for python with Harr object detection trainer
6. Program code for face detection written in Python Programming language

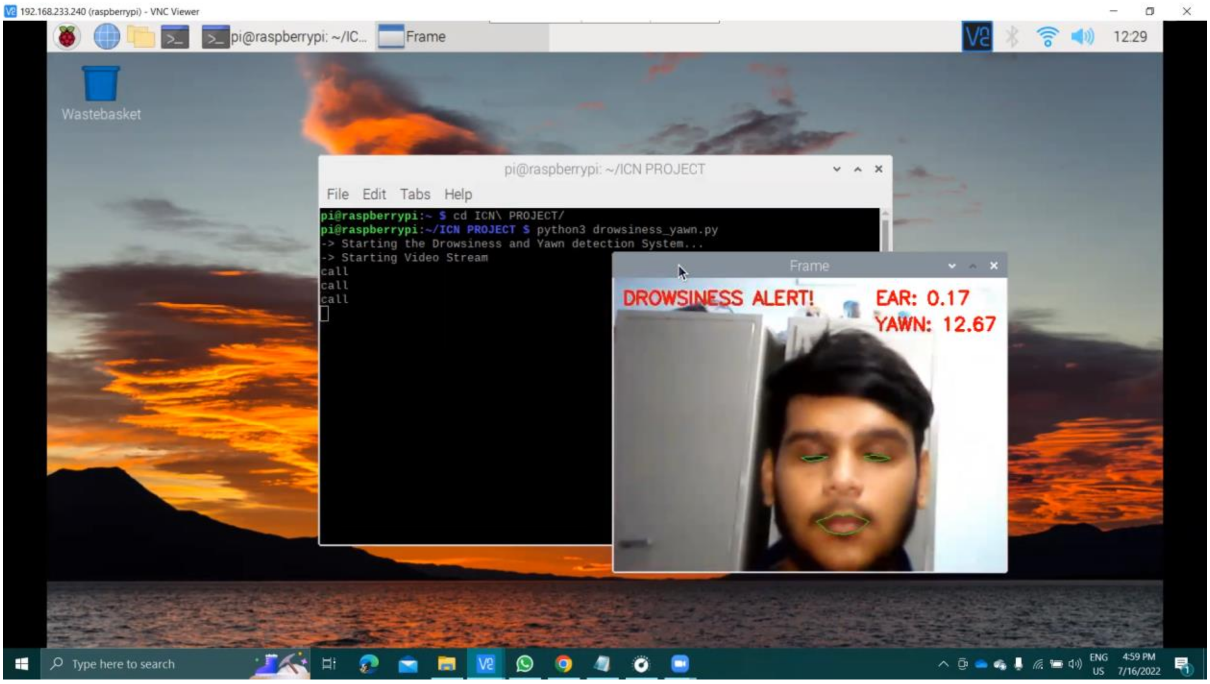
Now for the remaining libraries, you should not have any issues while installing them, as those are pretty straightforward. You need to install the following libraries in python 3:

- imutils
- scipy
- numpy
- argparse

5.2 Result

Prototype of drowsiness detection system was designed using RaspberryPi hard ware and coded in python language. It was tested with different subjects and different conditions and photo copies of the output were shown below.





Chapter 6

6.1 Conclusion

Implementation of drowsiness detection with RaspberryPi was done which includes the following steps: Successful runtime capturing of video with the camera.

Captured video was divided into frames and each frame was analyzed. Successful detection of the face followed by detection of the eye. If closure of the eye for successive frames were detected then it is classified as a drowsy condition else it is regarded as a normal blink and the loop of capturing the image and analyzing the state of the driver is carried out again and again.

In this implementation during the drowsy state, the eye is not surrounded by a circle or it is not detected and the corresponding message is shown. If the driver is not drowsy then the eye is identified by a circle and it prints 1 for every successful detection of an open eye.

Thank You