

# **DRIVER DROWSINESS DETECTION SYSTEM**

**A THESIS SUBMITTED IN PARTIAL  
FULFULMENT OF THE REQUIREMENTS  
FOR THE DEGREE OF**

**Bachelor in Technology  
In  
Electronics and Instrumentation Engineering**

**BY: CHANDRAPRAKASH SAHOO**

**ROLL NO: 112EI0563**

**GUIDED BY: Dr. Manish Okade**



**Department of Electronics and Communication Engineering**

**National Institute of Technology, Rourkela**



## National Institute of Technology, Rourkela

### DECLARATION

I declare that the project work with the title **“DRIVER DROWSINESS DETECTION SYSTEM”** is my own work done under **Dr. Manish Okade**, National Institute of Technology, Rourkela. I have gone through the rules of thesis writing provide by the institute and have followed all the instructions accordingly. This project work is being submitted in the fulfillment of the requirements for the degree of Bachelor of Technology in Electronics and Instrumentation Engineering at National Institute of Technology, Rourkela for the academic session 2012– 2016. All the performed experiments done under this projects and written in this thesis are properly performed by my own and has not been imitated from any other sources. If there is any reference made for theoretical purpose then due recognition is mentioned for the source of original publication.

**CHANDRA PRAKASH SAHOO (112EI0563)**



National Institute of Technology, Rourkela

May 16, 2016

## CERTIFICATE

This is to certify that the thesis titled “**DRIVER DROWSINESS DETECTION SYSTEM**” submitted by Chandra Prakash Sahoo (Roll no: 112EI0563) in partial fulfillment of the requirements for the award of Bachelor of Technology Degree in Electronics and Instrumentation engineering at National Institute of Technology, Rourkela is an authentic work carried out by him under my supervision and guidance.

## ACKNOWLEDGEMENT

It would be my immense pleasure to take this as a chance to represent my appreciation and earnest because of our regarded **Dr. Manish Okade** for the direction, knowledge, and bolster that he has given all through this work. My present work would never have been achievable without his guidance inputs and tutoring.

I would also like to thank every one of my friends, faculty team and all staff of prestigious Department of Electronics and Communication Engineering, N.I.T Rourkela for their amazing help all through our course of learn at this organization.

<b>CONTENTS:</b>	<b>Page</b>
Declaration.....	2
Certificate.....	3
Acknowledgement.....	4
Abstract.....	7
List of figures.....	8
➤ Chapter 1.....	9
1.1 Introduction.....	10
➤ Chapter 2.....	11
2.1 Drowsiness.....	11
2.2 Measures for detection of drowsiness.....	11
2.3 Proposed method.....	13
2.4 Drowsiness detection approach.....	14
➤ Chapter 3.....	17
3.1 Introduction.....	17
3.2 Object detection.....	17
3.3 Face detection.....	17
3.4 Eye detection.....	18
➤ Chapter 4.....	19
4.1 Introduction.....	19
4.2 Principal Component Analysis (PCA).....	19
4.3 Eigen face approach.....	19
4.3.1 Eigen value and Eigen vector.....	20
4.3.2 Face image representation.....	21
4.3.3 Mean and mean centered Image and Covariance matrix.....	21
4.3.4 Eigen face space.....	21
➤ Chapter 5.....	22
5.1 Introduction.....	22

5.2	Implementation.....	22
5.2.1	About RaspberryPi.....	22
5.2.2	Haar features.....	24
5.2.3	Integral Image.....	24
5.2.4	Adaboost.....	25
5.2.5	Cascading.....	26
5.2.6	Modification done in the algorithm.....	27
5.2.7	Result.....	28
➤	Chapter 6.....	32
6.1	Conclusion.....	32
6.2	Limitation.....	32
6.3	Future work.....	33

## **ABSTRACT**

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

Though there are several methods for measuring the drowsiness but this approach is completely non-intrusive which does not affect the driver in any way, hence giving the exact condition of the driver. 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.

<b>List of figures</b>	<b>Page</b>
Flow chart showing entire process of drowsiness detection system	14
Camera used for implementing drowsiness detection system	15
figure shows different components of RaspberryPi	23
Different features used for Haar cascade	24
figure shows integral image formation	25
Integral image calculation	25
Different stages of cascade classifier	27
Eye in open state with head position= straight. Circle around the eye	28
Eye in closed state with head position= straight. No circles around eyes.	28
Eye in open state with head position= Tilted (left).	29
Eye in closed state with head position= Tilted (left).	29
Eye in open state with head position= Tilted (Right).	30
Eye in closed state with head position= Tilted (Right).	30



# 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 say that around 30 percent of accidents are caused by fatigue of the driver. When a driver drives for more than a normal period for human then excessive fatigue is caused and also results in tiredness which drives the driver to a sleepy condition or loss of consciousness.

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

In chapter 1, in 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 of a drowsy condition and prevent accidents. Advantages and disadvantages corresponding to each and every system are 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 a face first. If a face is detected then the next task is to locate the eyes. After the positive result of detecting an eye the amount of closure of the 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 a face and detecting a drowsy condition is carried out.

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

In chapter 3, the theoretical base for designing the entire system is explained which includes Principal Component Analysis (PCA) and Eigen face approach. We know that the structure of a face is complex and multidimensional. A face needs great calculating methods and techniques for recognizing it. In this my approach will treat a face as a two dimensional structure and accordingly it should be recognized. Principal Component Analysis (PCA) is used for face recognition for this context. This idea involves the projection of face images onto that particular face space. Then we encode the variation or difference among the desired known faces. Eigen face decides and defines the face space. We represent these faces as eigen vectors. These vectors consist of all sets of faces. Cases of similarity with different features of our face appear like nose, eyes, lips etc.

The Eigen face approach uses the PCA for recognition of the images. The system performs by projecting pre extracted face image onto a set of face space that represents significant

variations among known face images. Eigen face approach includes Eigen Values and Eigen Vectors, Face Image Representation, Mean and Mean Centered Images, Covariance Matrix, Eigen Face Space.

In chapter 4, the implementation part is described. The hardware that is used for the entire system is Raspberry Pi. Hence a brief introduction on Raspberry Pi is given followed by theoretical approach in details on drowsiness detection system which includes Haar Cascade, forming Integral Image, Adaboost and Cascading. All those four above methods were used in finding the state of eyes and forming algorithm for that. Then proper code were written to implement the proposed algorithm. Raspberry Pi was set up properly for implementation. Several subjects were taken to record the response and working of the system. The opening of eyes were indicated by circular shapes. If drowsy state is detected then circle does not appear indicating the closure of eye or drowsy state of a driver. Results were shown with several photos with both eye opening and closing condition.

In chapter 6, drawbacks of the system were explained and the required future work to remove those drawbacks and to build a robust intelligent driver assistance system was emphasized. Finally the conclusive part includes the overall performance of the proposed and implemented system.

## **Chapter 2**

### **2.1 Drowsiness**

Drowsiness is defined as a decreased level of awareness portrayed by sleepiness and trouble in staying alert 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 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 been 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 in an edge of losing control over the vehicle prompting crash with other vehicle 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 factor. 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 project, 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 vehicle from lane position.
2. Sudden movement of steering wheels.
3. Pressure on acceleration paddles.

For each measures threshold values are decided which when crossed indicated that driver is drowsy.

**Advantages:**

1. It is noninvasive in nature.
2. Provides almost accurate result.

**Disadvantages:**

1. Vehicle based measures mostly affected by the geometry of road which sometimes unnecessarily activates the alarming system.
2. The driving style of the current driver needs to be learned and modeled for the system to be efficient.
3. The condition like micro sleeping which mostly happens in straight highways cannot be detected.

## **2. Physiological measures.**

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

ECG (electro cardiogram)

EMG (electromyogram)

EOG (electro oculogram)

EEG (electroencephalogram)

**Monitoring Heart Rate:** An ECG sensor can be installed in the steering wheel of a car to monitor a driver's pulse, which gives a sign of the driver's level of fatigue indirectly giving the state of drowsiness. Additionally the ECG sensor can be introduced in the back of the seat.

**Monitoring Brain Waves:** Special caps embedded with electrodes measures the brain waves to identify fatigue in drivers and report results in real time. Then each brain waves can be classified accordingly to identify drowsiness.

**Monitoring muscle fatigue:** As muscle fatigue is directly related to drowsiness. We know during fatigue the pressure on the steering wheel reduces and response of several muscle drastically reduces hence it can be measured by installation of pressure sensors at steering wheel or by measuring the muscle response with applied stimuli to detect the fatigue.

**Monitoring eye movements:** Invasive measurement of eye movement and eye closure can be done by using electro oculogram but it will be very uncomfortable for the driver to deal with.

Though this method gives the most accurate results regarding drowsiness. But it requires placement of several electrodes to be placed on head, chest and face which is not at all a convenient and annoying for a driver. Also they need to be very carefully placed on respective places for perfect result.

### **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**

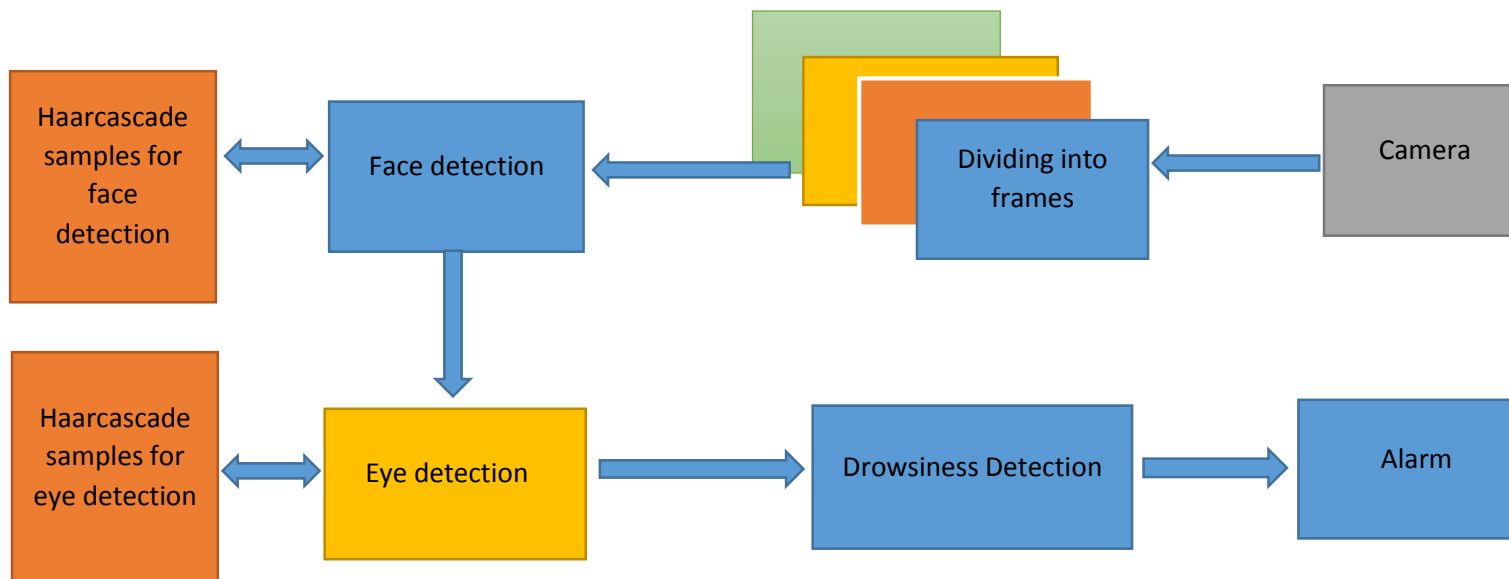
Among all these four strategies, the most precise technique depends on human physiological measures [1]. This procedure is executed in two ways: measuring changes in physiological signs, for example, brain waves, heart rate, and eye flickering; and measuring physical changes, for example, sagging posture, inclining of the driver's head and the open/shut conditions of the eyes [1]. In spite of the fact that this procedure is most precise, it is not reasonable, since detecting electrodes would need to be put straightforward 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 be mostly focusing on 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 condition does not affect

this system. The case of micro nap is also detected according 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

**Flowchart of the proposed system**



**Figure 1: Flow chart showing entire process of drowsiness detection system**

## **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. This paper is only focused on the applying the proposed mechanism only on single frame. The used camera is a low cost web camera with a frame rate of 30 fps in VGA mode. Logitech Camera is used for this process is shown in figure 2.



*Figure 2: Camera used for implementing drowsiness detection system*

### **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.

### **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 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 Open Cv 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  $\Delta$  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 lots 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 method 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. In the beginning of this chapter Principal Component Analysis (PCA) method was described. Latter through this Eigen face approach is explained which includes mathematical description of its subparts such as Eigen Values and Eigen Vectors, Face Image Representation, Mean and Mean Centered Images, Covariance Matrix and Eigen Face Space.

### **4.2 Principal Component Analysis (PCA)**

Principal component analysis (PCA) was invented in 1901 by Karl Pearson. If the resulted data is repeated again and again or has redundancy the PCA helps in reducing this redundancy. PCA basically removes the variables to reduce redundancy. So after reduction of variables we will get less variables named as Principal Components. Principal components will generally represent all the variables present in the obtained variable. But it only reduction of variables does not solve the purpose. Main Problem appears when we try to achieve face recognition in a more and high dimensional space. The main objective of PCA is to decrease the no of dimension as well as retain more and more possible variation in the given data set. But we know that reduction in dimension results in information loss as information are directly linked with dimension. Hence we can overcome the problem of data loss by choosing the best principal components as main principal components determines the low dimension. Though use of PCA has many advantages but mostly it is used for eigen face approach. In eigen face approach the reduction of size of the data base is achieved for recognizing the test images. The obtained images are stored in the data base in vector form which are also called feature vectors. And these are found out from set of Eigen face obtained by projecting it over trained image. So basically PCA is used for Eigen face approach for the reduction of dimensionality with our causing the loss of data.

### **4.3 Eigen face approach**

Eigen face approach for face recognition is very efficient and helpful because of its speed of operation simplicity in using and capability of learning. In computer vision face detection is done by use of eigen face which are basically set of eigen vectors. This approach is basically an appearance based approach which does face recognition by capturing the variation in a set of face images and this information is used for comparison and encoding of each individual faces in proper manner. What we mean by eigen faces is that they are Principal components of distributed faces which are represented in the form of covariance matrix of set of faces. In this method a face image is represented in the form of one dimensional matrix. We know we can represent a face in two dimensional form of pixels as  $N \times N$  matrix in  $N^2$  dimension space. These  $N \times N$  matrix is shifted to the form of row matrix. Many work on this were already done

but it has ignored the fact of face stimulus which assumes that the given predefined measurements on face recognition are important and adequate. Which means that coding and encoding of available face images probably give information of face images which point out the important significant features. But a chance is there that the obtained features may or may not be related to the known and required facial feature like nose, eyes, lips, hair etc. So the extraction of required information from a face image is required. After extraction is done we encode it with high efficiency and the result is compared with a database of faces encoded in the same fashion. For this purpose we capture the variation with a collection of face images which is a very simple approach for the extraction of the information content. The next step is to find the Principal Component of the face distribution or from the obtained covariance matrix the Eigen vectors of the set of face images can be found out. Every row of image is considered as a vector stacked one after another in a single row which helps in displaying the Eigen vector as a sort of face. A linear combination of face images are taken to represent each face images. We find that the sum total of all expected eigen faces is decided by total number of given input images in the prepared set. An approximation can be done for faces by the use of Eigen face for those having large eigen values which set the most variance in in case of available set of images. To increase the computational efficiency use of fewer Eigen face is done.

#### **4.3.1 Eigen values and eigen vectors:**

In linear algebra, a linear equation in matrix form is represented by  $Ax = D$ .

The eigenvectors of a linear operator are non-zero vectors which, when operated by the operator. The result of this is a scalar multiple of them. For the eigen vector  $X$  the obtained scalar called eigen value ( $\lambda$ ). A vector which is paralleled by linear transformation is called an Eigen vector. It is one of the property of matrix. When we calculate a matrix on it then the magnitude of the vector is changed. The direction of vector remains as it is. So we define as  $Ax = \lambda x$ , where  $A$  is represented as a vector function. Then transforming the RHS part and writing it as  $(A - \lambda I)x = 0$ , where  $I$  is called the identity matrix. The above form is a homogeneous equation and is fundamental part of linear algebra. Existence of non-trivial solution is decided by considering that if and only if  $\text{Det}(A - \lambda I) = 0$ , where  $\text{Det}$  represents determinant. When it is evaluated we deal with the polynomial of degree  $n$ . This is known as the characteristic polynomial of  $A$ . If we represent the dimension of  $A$  by  $N \times N$  then the solutions results in  $n$  roots of the characteristic polynomial. So it gives  $n$  Eigen values of  $A$  which satisfy the  $Ax_i = \lambda_i x_i$ , where  $i = 1, 2, 3, \dots, n$ . If the obtained eigen values are all distinct then we get  $n$  associated linearly independent eigen vectors with unique directions.

#### **4.3.2 Face Image Representation**

In this approach we represent set of let's say  $m$  images of each having size  $N \times N$ . This is done by vectors of size  $N^2$ . We represent each face  $\Gamma_1, \Gamma_2, \Gamma_3, \dots, \Gamma_n$ . All those obtained feature vectors are stored in the matrix with size  $N \times N$ . One example is shown below which describes the entire process. For example:

$$\begin{bmatrix} 3 & 7 \\ 6 & 5 \end{bmatrix} = \begin{bmatrix} 3 \\ 7 \\ 6 \\ 5 \end{bmatrix}$$

### 4.3.3 Mean and mean centered Image and Covariance matrix

We calculate the average face by

$$\Psi = \sum_{n=1}^{\infty} \Gamma_n$$

Then we find the difference of each face from their average face which is

$$\Phi_i = \Gamma_i - \Psi$$

We can construct a covariance as mentioned below.

$C = AA^T$ , where  $A = [\Phi_1, \Phi_2, \dots, \Phi_m]$  of size  $N^2 \times N^2$ .

As we can see that the size of covariance matrix will be  $N^2 \times N^2$  which is huge actually and we need to find the Eigen vectors for the covariance matrix. But the large size make it time consuming and tedious. To encounter this problem we go for calculating  $A^T A$ .

Now let's consider the eigenvectors  $V_i$  of  $A^T A$  such that  $A^T A X_i = \lambda_i X_i$ .

The eigenvectors  $V_i$  of  $A^T A$  are  $X_1 \dots X_n$ . Now for simplifying we multiply the above equation with  $A$  both sides and we get

$$AA^T A X_i = A \lambda_i X_i$$

$$AA^T (A X_i) = \lambda_i (A X_i)$$

From above we clearly see that Eigen vectors responding to  $AA^T$  is now firmly computed by reduction in dimension where  $AX_i$  is the Eigen vector and  $\lambda_i$  is the Eigen value.

### 4.3.4 Eigen face space

Let's say we have a covariance matrix  $AA^T$ . So the eigen vectors corresponding to that matrix which is denoted by  $U_i$  where  $U_i$  represents facial images. Those eigen faces basically look like ghostly. Only those eigen vectors are accepted which corresponds to Eigen face in the face space and discarded faces are faces are having eigen values zero. This method helps to reduce the Eigen face to a great extent. Rank of Eigen faces are decided according to their usefulness to characterize the variation among the images. Here we project the face image into the face space by  $\Omega_k = U^T(\Gamma_k - \Psi)$ ;  $k=1, \dots, M$ , where  $(\Gamma_k - \Psi)$  is the mean centered image. Hence we obtain the projection of each image as  $\Omega_1$  for projection of image1 and  $\Omega_2$  for projection of image2 and hence forth.

## Chapter 5

### 5.1 Introduction

Chapter 5 includes the implementation of the drowsiness detection system with the hardware. The hardware 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.

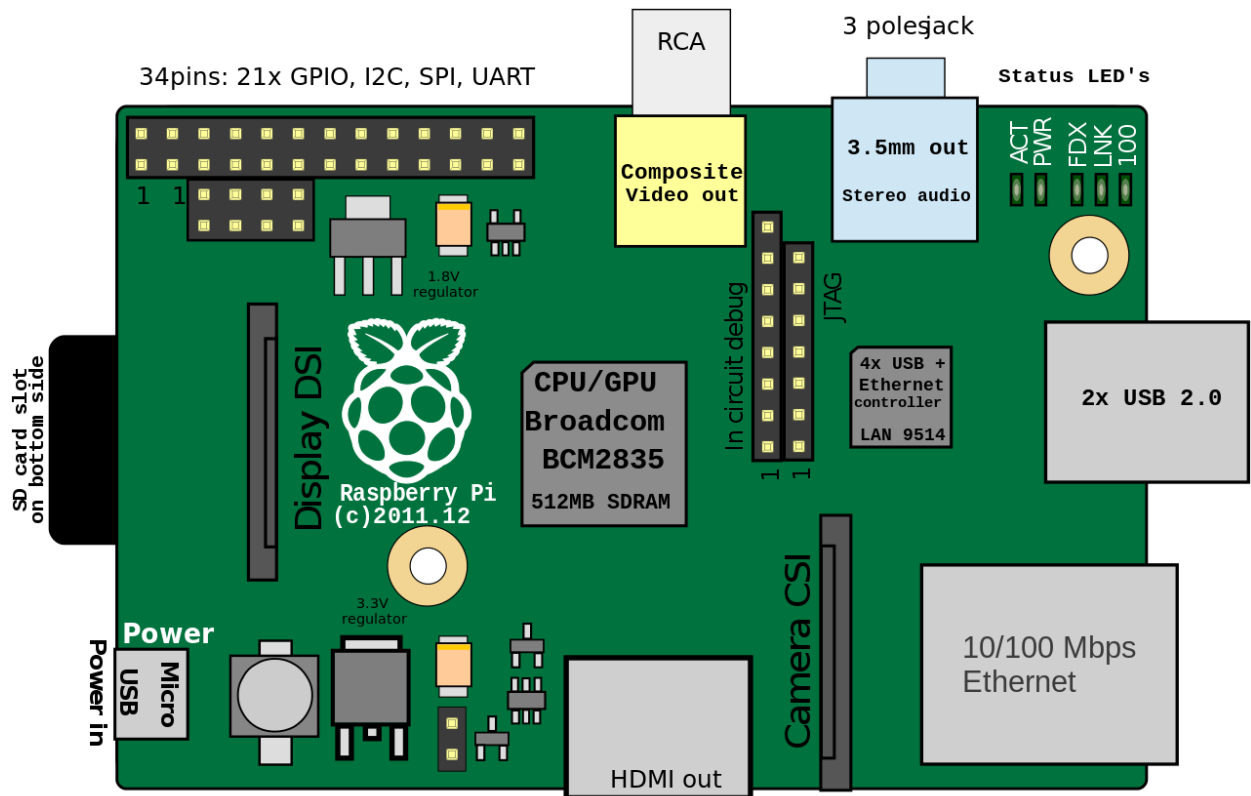
### 5.2 Implementation

#### 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.



**Figure 3:** figure shows different components of RaspberryPi

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.

The face detection method used in OpenCv is developed in 2001 by Paul Viola and Michael Jones, very well referred to as the Viola-Jones method. Though this method can be used for several objects but most specifically here it is used for face and eye detection in real time.

**Viola-Jones algorithm has four stages:**

1. Haar Feature Selection

2. Creating an Integral Image
3. Adaboost Training
4. Cascading Classifiers

### 5.2.2 Haar features

**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. So basically we move those features throughout our face to find the output of each feature.

For example:

All faces share some similar properties

1. The eyes region is darker than the upper-cheeks.
2. The nose bridge region is brighter than the eyes.

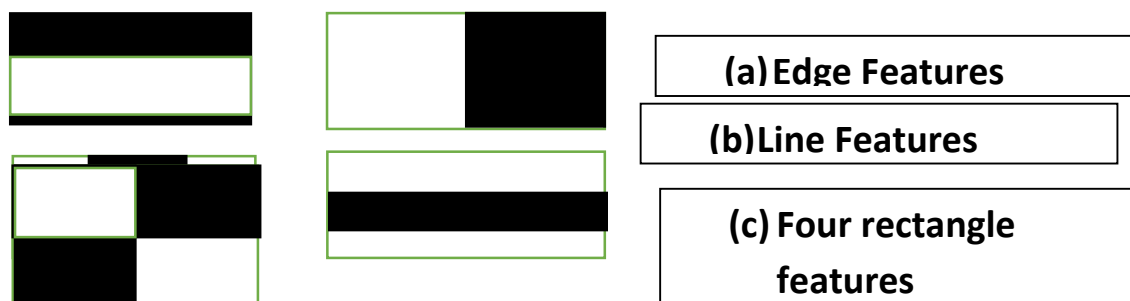
So this features of face are used for developing haar like features.

Each feature is related to a special location in the face.

Output of Rectangle features:

We will move the related kind of rectangle throughout the face to get different values.

1. Value =  $\sum$  (pixels in black area) -  $\sum$  (pixels in white area).
2. Three types: two-, three-, four-rectangles, Viola and Jones used two-rectangle features.



*Figure 4: Different features used for Haar cascade*

### 5.2.3 Integral Image



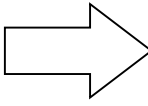
We know each point of an image is represented by a pixel value. As we so we need to know the output of applied Haar features so we need to find the sum of pixel value of all those area and solve the summation. But this is a huge task. To reduce the number of calculation concept of INTEGRAL IMAGE is introduced.

Definition of Integral Image:

Basically Integral image is a matrix same as size of the window. The integral image at location (x, y) is the sum of the pixels above and to the left of (x, y).

For example

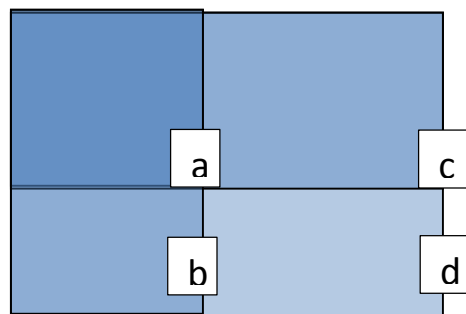
0	1	1	1
1	2	2	3
1	2	1	1
1	3	1	0



0	1	2	3
1	4	7	11
2	7	11	16
3	11	16	21

**Figure 5:** figure shows integral image formation

Here the pixel value of each box is modified by sum of all those box left and above it so that we can use a formula mentioned below to get the output of Haar features with much less calculation reducing the time of calculation.



For example the integral sum of inside rectangle can be computed as:

$$ii(d) + ii(a) - ii(b) - ii(c)$$

Where  $ii$  stands for integral image value.

**Figure 6:** Integral image calculation

Hence if we want to calculate the pixel value of a rectangle we can do so by just taking four points from integral image as above.

#### 5.2.4 AdaBoost

Adaboost stands for “Adaptive” boost. Here we construct a strong classifier as linear combination of weak classifier as there are so many features which are absolutely invalid in finding the facial features. It can be formulated as below:

$$F(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x) + \alpha_3 f_3(x) + \dots$$

Strong Classifier

Image

Weight

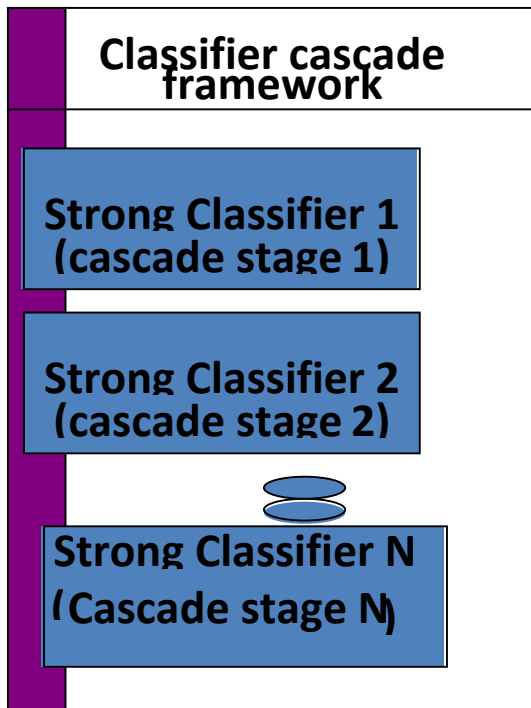
Weak classifier

Features of weak classifiers:

Each single rectangle is regarded as a simple weak classifier. Each weak classifier is assigned a weigh function as per its importance of position. Finally the strong classifier is formed by their linear combination.

#### 5.2.5 Cascade

After going through Adaboost stage now let's say we have 600 no of strong classifiers. So to detect if a frame contains a face or not: Instead of applying all the 600 features on a window, group the features into different stages of classifiers and apply one-by-one. If a window fails the first stage, discard it. We don't consider remaining features on it. If it passes, apply the second stage of features and continue the process. The window which passes all stages is a face region.



*Figure 7: Different stages of cascade classifier*

## 5.2.6 Modification in the algorithm

### Modification in face detection:

For detection of face, loading of cascade file is done. Then the requirement is to pass on the captured frame to a function which does detection of edge. After this process almost all the possible type objects may get detected corresponding to different sizes. Hence the task of reduction of reduction of processing amount comes here. To achieve this instead of Detecting all the objects present in the frame because we know that our required object is face and in most case it occupies almost all the portion of the captured frame. So we can modify the algorithm to detect only in that fashion.

### Modification in eye detection:

For eye detection as the amount of processing will be very high if we apply the features to all the portion of the face. Hence to avoid this situation we should be interested only in those portion of face where we are certain that eye exist there. We set our work region for finding the eye by taking the following facts.

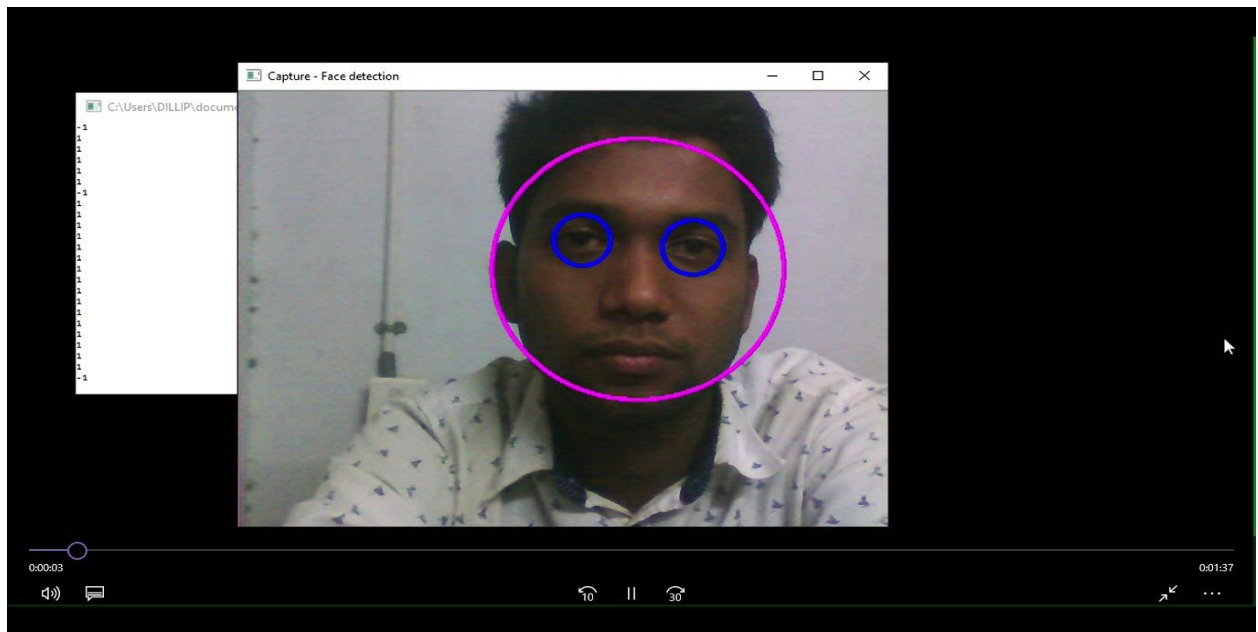
1. Only upper parts of the face contain the maximum probability of finding an eye.
2. The place of occurrence of eyes are a few pixel below the fore head.

### Modification in colour selection

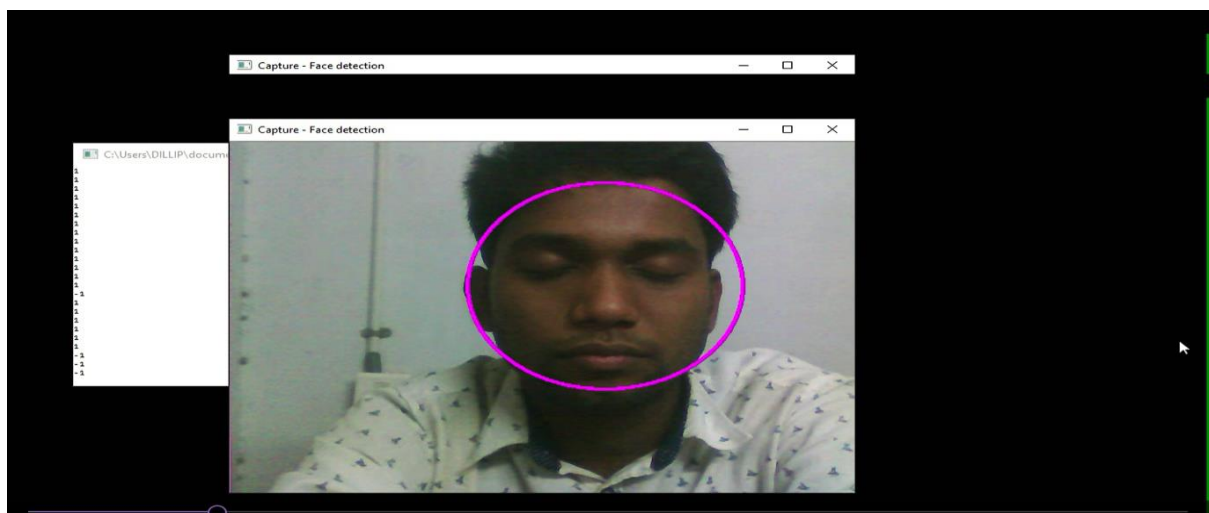
Now for drowsiness detection instead of using the colored image, the image is converted to gray scale to reduce the no of channel parameters which helps to increase the speed of calculation of the classifiers.

### 5.2.7 Result

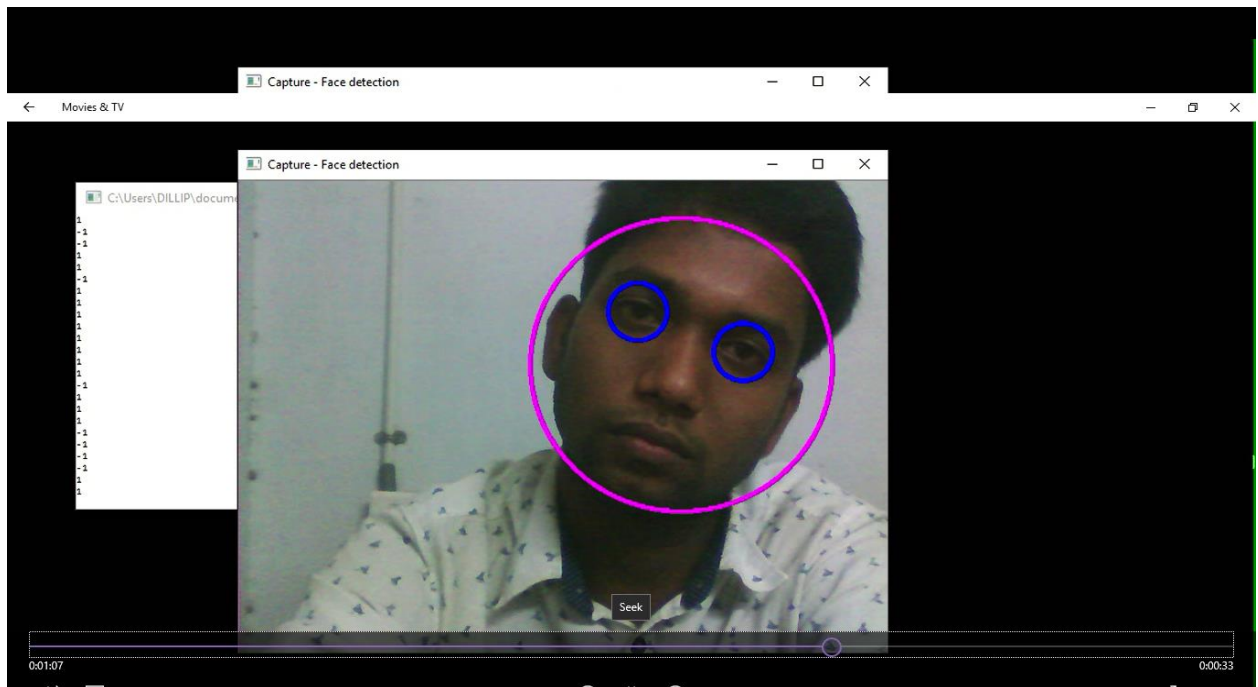
Prototype of drowsiness detection system was designed using RaspberryPi hard ware and coded in python language. It was tested with different subjects and different condition like straight and tilted head and photo copy of the output was shown below.



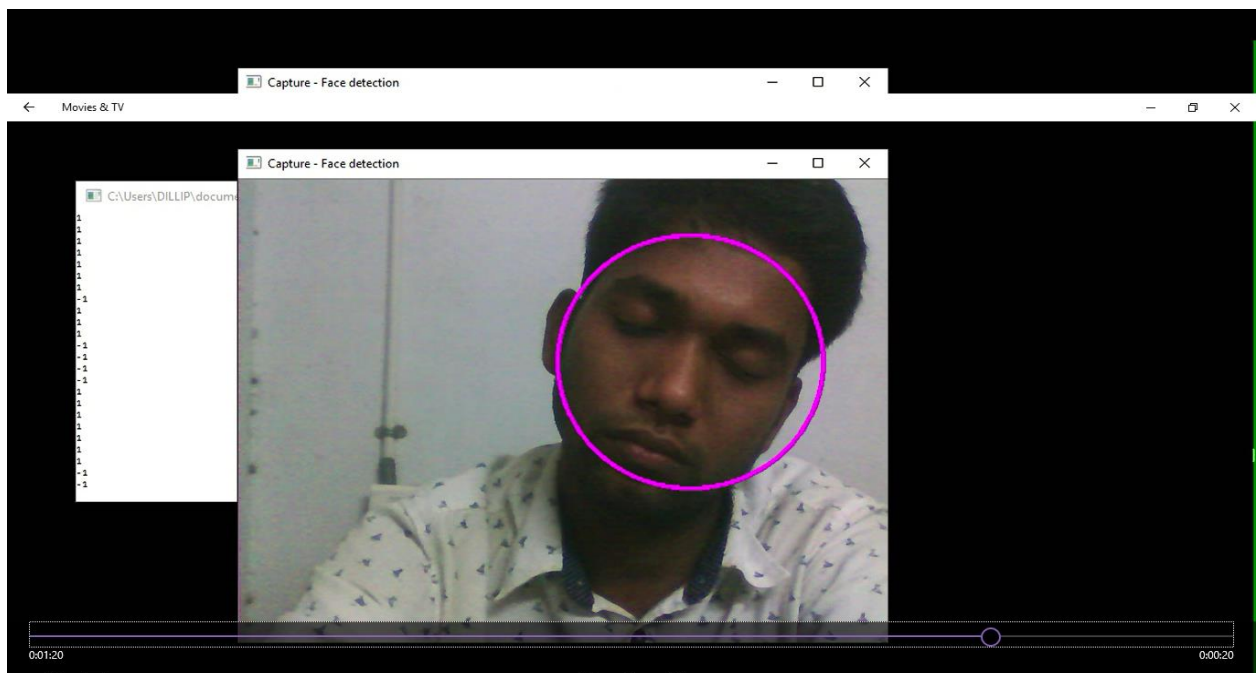
*Figure 8: Eye in open state with head position= straight. Circle around the eye.*



*Figure 9: Eye in closed state with head position= straight. No circles around eyes.*



*Figure 10: Eye in open state with head position= Tilted (left).*



*Figure 11: Eye in closed state with head position= Tilted (left).*



The result is obtained by taking several position of head like straight, tilted (Right), tilted (Left) etc. We can observe that when the eyes are open circles appears around eye indicating open state of eye. When eyes are closed circle disappears indicating closed state of eye. In both the cases face detection occurs which is shown by a pink colored circle. In the side window the output parallels with one and zero according to the fact that eye is opened or closed respectively.

## Chapter 6

### 6.1 Conclusion

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

Captured video was divided into frames and each frames were analyzed. Successful detection of face followed by detection of eye. If closure of eye for successive frames were detected then it is classified as drowsy condition else it is regarded as normal blink and the loop of capturing image and analyzing the state of driver is carried out again and again. In this implementation during the drowsy state the eye is not surrounded by circle or it is not detected and corresponding message is shown. If the driver is not drowsy then eye is identified by a circle and it prints 1 for every successful detection of open eye.

### 6.2 Limitation

**Dependence on ambient light:** The model developed for this purpose strongly depends on the ambient light condition. As our algorithm considers the eye sight as a dark region when it is closed and brighter region when it is open so if the ambient condition affects such that there may be possibility of brighter and darker condition depending on light source then it causes error in the result. Also this model depends on certain minimum level of light condition otherwise it becomes very difficult to detect. To avoid this error we can use either LED light for better detection or we can use an infrared camera.

**Distance of camera from driver face:** For best result we have assumed and designed the code according to the fact that the distance between camera and face should be nearly 100 cm. Hence the designed set up output may vary from vehicle to vehicle as different vehicle have different types of seat lengths.

**Processor speed of hardware:** We have used RaspberryPi for implementation. The processor speed of RaspberryPi is 700 MHz. So this speed of processor is not compete enough to do video processing. Hence processor with very high speed is needed which will ultimately increase the cost of the product.

**Use of spectacles:** In case the user uses spectacle then it is difficult to detect the state of the eye. As it hugely depends on light hence reflection of spectacles may give the output for a closed eye as opened eye. Hence for this purpose the closeness of eye to the camera is required to avoid light.

**Multiple face problem:** If multiple face arise in the window then the camera may detect more number of faces undesired output may appear. Because of different condition of different faces. So we need to make sure that only the driver face come within the range of the camera. Also the speed of detection reduces because of operation on multiple faces.



### **6.3 Future work**

Our model is designed for detection of drowsy state of eye and give an alert signal or warning may be in the form of audio or any other means. But the response of driver after being warned may not be sufficient enough to stop causing the accident meaning that if the driver is slow in responding towards the warning signal then accident may occur. Hence to avoid this we can design and fit a motor driven system and synchronize it with the warning signal so that the vehicle will slow down after getting the warning signal automatically. Also we can avoid the use of RaspberryPi which is not so fast enough for video processing by choosing our own mobile phone as the hardware. This can be done by developing a proper mobile application which will perform the same work as RaspberryPi and response will be faster and effective.

## **BIBLIOGRAPHY**

- [1]. W. Zhao, R. Chellappa, P.J. Phillips, and A. Rosenfeld, "Face Recognition: A Literature Survey," ACM Computing Surveys, vol. 35, pp. 399-459, 2003.
- [2] M. H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: A survey," IEEE Trans. Pattern Anal. Mach. Intell., vol.24, no.1, , Jan. 2002, pp. 34–58.
- [3]. Nan-Ning Zheng, Shuming Tang, Hong Cheng and Qing Li, Guanpi Lai and Fei-Yue Wang, "Toward Intelligent Driver-Assistance and Safety Warning Systems", Intelligent Transportation System, IEEE 2004.
- [4]. Christian Scharfenberger, Samarjit Chakraborty, John Zelek and David Clausi", "Anti-Trap Protection for an Intelligent Smart Car Door System", 15th International IEEE Conference on Intelligent Transportation System, Anchorage, Alaska, USA, September 16-19, 2012.
- [5]. K. C. Yow and, R. Cipolla, "Feature-based human face detection," Image Vision Comput., vol.15, no.9, 1997, pp.713–735.
- [6] An Analysis of Viola Jones algorithm for face detection by Yi-Quin Wang, University of Malaysia Phang, 2014, pp: 15-20.
- [7] Implementation of Viola Jones Algorithm by Ole Helvig Jensen, university of Denmark, 2008, pp: 20-36.