Driver drowsiness detection using behavioural measures and machine learning techniques

CS4090 Project Final Report

Submitted by

Akansh Kumar	**B150398CS**
Meshram Rohit Ajit	**B150895CS**
Patan Daen Khan	**B150308CS**
Tripathi Sandilya Ashutosh	**B150640CS**

Under the Guidance of
Dr. Jay Prakash

Department of Computer Science and Engineering
National Institute of Technology Calicut
Calicut, Kerala, India - 673 601

April 29, 2019

NATIONAL INSTITUTE OF TECHNOLOGY CALICUT, KERALA, INDIA - 673 601

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

तमसो मा ज्योतिर्गमय

2019

CERTIFICATE

Certified that this is a bonafide record of the project work titled

DRIVER DROWSINESS DETECTION USING BEHAVIOURAL MEASURES AND MACHINE LEARNING TECHNIQUES

done by

Akansh Kumar

Meshram Rohit Ajit

Patan Daen Khan

Tripathi Sandilya Ashutosh

of eighth semester B. Tech in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering of the National Institute of Technology Calicut

Project Guide	**Head of Department**
Dr. Jay Prakash	Dr. Saleena N
Assistant Professor	Associate Professor

DECLARATION

I hereby declare that the project titled, **Driver drowsiness detection using behavioural measures and machine learning techniques**, is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or any other institute of higher learning, except where due acknowledgement and reference has been made in the text.

Place :	Signature :
Date :	Name :
	Reg. No. :

Abstract

Drowsiness detection system is regarded as an effective tool to reduce the number of road accidents. This project proposes a non-intrusive approach for detecting drowsiness in drivers, using Computer Vision. Developing various technologies for monitoring and preventing drowsiness while driving is a major trend and challenge in the domain of accident avoidance systems. Haar face detection algorithm is used to capture frames of image as input and then the detected face as output. Haar feature based Adaboost algorithm is used to extract the facial region from the image.Facial landmarks are used to extract the draw contours around the eyes and mouth of the driver. EAR and MAR values are calculated and if their values exceed a certain threshold, the user/driver is alerted to be in a state of drowsiness.Head tilt is estimated to determine the drivers attention on the road.

ACKNOWLEDGEMENT

We would like to thank our guide Dr Jay Prakash for always supporting and mentoring us. My thanks and appreciations also go to my colleagues in developing the project. It would not have been possible without the kind support and help of these people.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Road accident is a global tragedy with number of cases increasing year by year.Owing to the bad infrastructure and dangerous driving habits, Most deaths around 105,000 per year occur in India. Around 20 percent of traffic accidents and up to 25 percent of serious accidents occur because of Driver's diminished vigilance. The increasing number of accidents due to a drivers vigilance level diminishing has become a grave problem for us.

The algorithm is coded on OpenCV platform in Linux environment. The parameters considered to detect drowsiness are face and eye detection, blinking, eye closure and head tilt . The algorithm is Haar trained to detect the face. Once the face is detected, the facial landmarks position around the eyes is determined. The mean eye landmarks distance is calculated and thus the eye state is determined from that distance. Eyelid closure/blink/gaze is detected using the values obtained from each of the incoming frames.

Facial region is detected by Viola Jones Algorithm. Main emphasis is paid to processing of data and faster detection. Drowsiness of the driver will be found by finding whether the eyes are shut over subsequent frames and similarly it can be found if the driver is yawning by observing the mouth region.Yawning and closing of eyes may suggest that the driver is drowsy and hence sound the alarm.

Chapter 2

Literature Survey

Attempts to detect drowsiness using EEG(electroencephalogram) have been made. To detect multiple features in a subject, ECG and EMG also have been used in parallel .A lot of feature based matching methods and template based had been developed in the past. A number of parameters can be used to detect drowsiness.

Using eye related parameters is one of the ways to detect drowsiness.Repeated experiments reveal that the most valid measure of loss of alertness among drivers is the percentage of eyelid closure over the pupil over times. In the past, Blink detection by analyzing the light pupils have also come up . Percentage of eyelid closure is one of the chosen parameters to detect drowsiness in a driver[1]

Vehicle behavior analysis in driver drowsiness situation measures vehicle behaviors such as speed,position and exacerbation angle.They can be considered as non-intrusive but they have multiple limitations like type of the the driving conditions,car, etc. Further, it requires preparation and special equipment and can be expensive and not practical [2].

Computer vision analysis benefits from the dynamic behavior of the human face and eye since they have a high degree of variability. Face detection is considered to be a difficult problem in computer vision research, whereas the eyes can be said as relatively stable feature on the face in comparison with other facial features.

Using Support Vector Machine(SVM), various states of drowsiness have been detected in the past.OpenCV is used to implement computer vision algorithms that gives optimum results. Haar Classifiers based on Viola Jones algorithm is one of the widely accepted way to detect various objects. Multilayer Perceptron algorithm was popularly used in a number of detection methods for various entities [1].

Nowadays, robust real time facial landmark detectors capture most of the characteristic points on a human face image, including eyelids and eye corners. These are trained on in-the-wild datasets and thus are robust to varying facial expressions,illumination and moderate non-frontal head rotations. In real time applications like biometric and surveillance, the pose variations and camera limitations make the detection of human faces in feature space more complex than that of frontal faces. It further complicates the problem of robust face detection. Paul Viola and Michael Jones presented a fast and robust method for face detection which is 15 times quicker than any technique at the time of release with 17 frames per second giving 95 percent accuracy [3]

High detection rates are achieved by Viola Jones. Images get processed faster and rapidly.Real time systems use it mostly because only on the present single grayscale image [4].The main sections of viola jones is listed below:

- Integral Image allowing speedy evaluation of the features.
- Classifier function built on a few features.
- Combining the classifiers in a cascade structure increases the speed of the detector as it focuses on regions of interest.

Speed, efficiency and accuracy are ensured by Viola-Jones Algorithm.It is used as the basis of our design. Algorithm looks for specific Haar feature of a face, if this feature found algorithm pass the candidate to the next stage.

2.1 Integral Image

$$P(x, y) = \sum_{x' < x, y' < y} i(x', y')$$

$$P1 = A,\ P2 = A + B,\ P3 = A + C,\ P4 = A + B + C + D$$

$$P1 + P4 - P2 - P3 = A + A + B + C + D - A - B - A - C = D$$

The use of integral image representation helps in faster detection of haar like features. Haar like features in an image are too large. In a sub image of size 24x24 the number of features contained is 180,000.

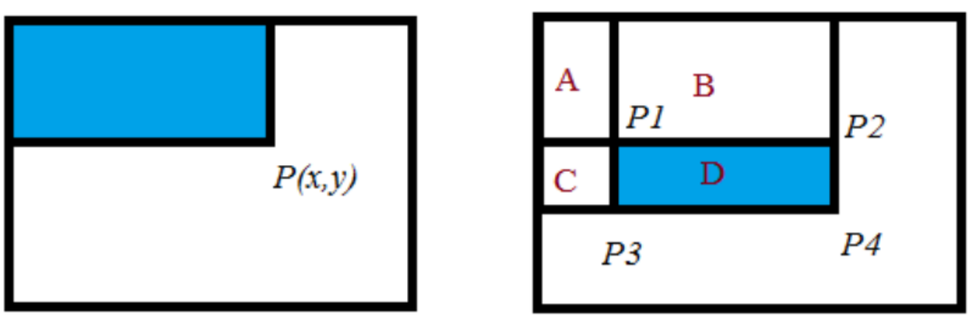

Figure 2.1: Integral image at location of x, y contains the sum of the pixel values above and left of x, y, [9] inclusive:

2.2 Haar Features

Similarities exist in human face. We use this fact for haar features. They are composed of two or three rectangles. These features are applied on face candidate to find out whether face is present. These features have a value and it can be found by calculating the area of each rectangle and then adding the result.We use this concept to easily find out the area of Rectangle.

Haar Features present in the current stage are searched in the image. Adaboost generates the size , weight the even the features themselves.[5] Taking the area of each rectangle, multiplying by their respective weights, and then adding the results gives us value of each feature. Use of the integral image leads to finding the rectangle's area. The addition of each pixel above and to the left of it using the integral image gives the coordinate of any corner of a rectangle[4]

Haar like features used over raw pixel values reduce/increase the in-class/out-of-class variability, and that makes the classification easy. Fig 2.2 shows the commonly used rectangular features.

2.3 Classifier

Value of Haar feature is calculated using the integral of rectangle. Many Haar classifiers make a stage. A stage accumulator sums all the Haar classifier results and a stage comparator compares against a stage threshold.Weak classifiers reject the region where the probability of finding face is less. The features and their threshold are selected in training phase. Training is done with AdaBoost algorithm. The threshold is a constant returned from the AdaBoost algorithm. Haar features, Depending on the parameters of the training data individual stages can vary in number. Many Haar Feature classifiers make up a stage. [4]

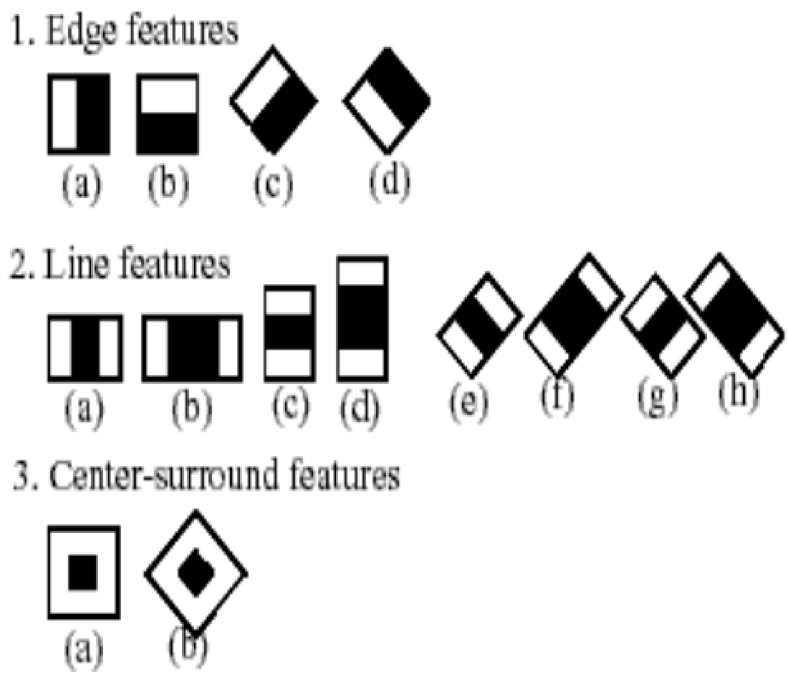

Figure 2.2: Haar Rectangular Features [5]

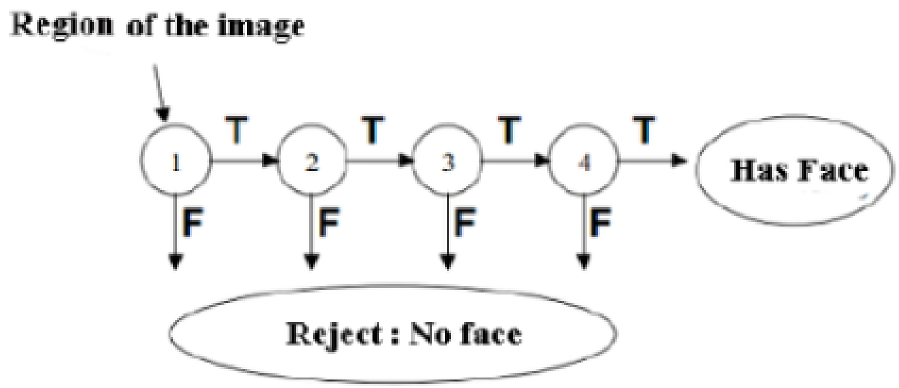

Figure 2.3: Cascades For Face Detection [7]

2.4 Cascade

The cascade discards images by making each subsequent stage more hard for a candidate image pass. A Frame or an image exits the cascade if they pass all stages or fail any stage.[4] If an image passes all the stages, that means a face is detected.

2.5 Facial Landmarks

To find the landmarks on the faces, a shape estimator is implemented in the dlib library based on the paper [12]. The estimator gives 68 landmark points,which includes the corner of the eyes, the tip of the nose, etc.After estimating positions of the facial landmarks, The indexes of the 68 coordinates we achieved can be visualized from the figure 2.4.

We intend to use facial landmark detectors to localize the eyes and eyelid contours. It is an inbuilt Histogram of Gradients Support Vector Machine classifier used to determine the locations of 68 different (x,y) coordinates that

Figure 2.4: 68 facial landmark points on face [12]

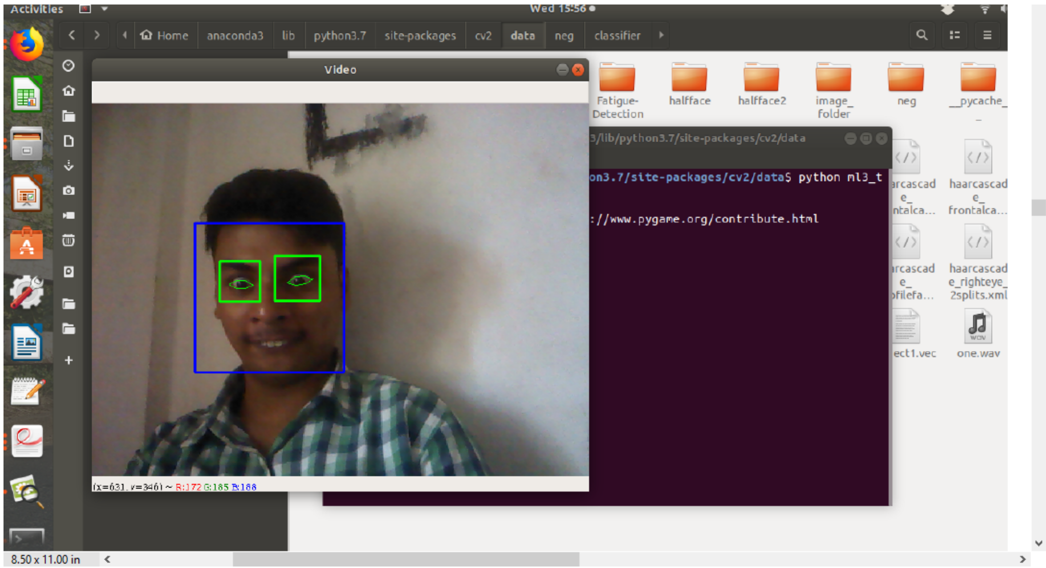

Figure 2.5: Facial Landmark around eyes

map to facial structures.

A single scalar quantity that reflects a level of the eye opening is derived from the landmarks.We determine the eye aspect ratio (EAR) that is used to measure eye opening state. After determining the eye and mouth landmarks, EAR and mouth aspect ratio(MAR) are determined by calculating the Euclidean distance between the landmarks using SciPy library.As per-frame EAR may not necessarily recognize the eye blinks correctly, a classifier that takes a larger temporal window of a frame into account is trained.

2.6 Eye Aspect Ratio (EAR)

For every video frame, the eye landmarks are determined. The eye aspect ratio (EAR) between height and width of the eye is calculated, where p1,...,p6 are 2D landmark locations. The EAR is mostly constant when an eye is open and approaches zero when closing an eye. Aspect ratio of the open eye has a small variance among individuals and it is fully invariant to a uniform

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

Figure 2.6: Vector formula for EAR.

scaling of the image and in-plane rotation of the face.the EAR of both eyes is averaged since eye blinking is performed by both eyes synchronously.[10].

2.7 Mouth Aspect Ratio (MAR)

Aspect ratio is an image projection attribute that describes the proportional relationship between the height and width of an image. With contour of the mouth, it is relatively easy to decide whether a person is drowsy or not by checking the size of the mouth. If the height is greater than a certain threshold, which means a person is yawning.

2.8 Head Tilt Estimation

In Drowsiness detection scenario head pose is an important factor. Here we use head tilt estimation as one of the factors to predict alertness/drowsiness of driver. It is based on perspective-n-point problem.

The relationship between a point on the 3-D head model and a point in the 2-D image is expressed below

Where,(X, Y, Z) denote the coordinates of a 3-D points in the world coordinate space, and (u,v) denote the coordinates of the projection point in pixels. The main idea is to determine the correspondences between 2D image features and points on the 3D model curve.

$$s\begin{bmatrix}u\\v\\1\end{bmatrix}=\begin{bmatrix}f_x & 0 & c_x\\0 & f_y & c_y\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}r_{11} & r_{12} & r_{13} & t_1\\r_{21} & r_{23} & r_{23} & t_2\\r_{31} & r_{32} & r_{33} & t_3\end{bmatrix}\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}$$

Chapter 3

Problem Definition

The overall human facial landmarks while working can be detected, under vehicular driving scenarios. Regular and real time monitoring of people/drivers is carried out. The goal of the project is analysis by regularly monitoring the driver/user's concentration/attention/drowsiness, etc. while driving and based on the observation sound an alarm .

Chapter 4

Methodology

4.1 Design

4.1.1 Overview of Design

Speedy detection and processing of data is our main aim. Frame count of closed eyes is kept. If the frame count goes above the threshold, then a alert message is displayed showing that the driver is feeling drowsy. First image is collected.To detect the faces in each individual frame our trained Haar Cascade file(classifier/cascade.xml) is used. We move on next frame if no face is found. [6] The region of interest marked after facial region are detected ,contains the eyes and mouth. ROI substantially reduces the computational requirements and eases the way for facial landmarks.

After finding the face and eye our progress is dependent on facial features like the corner of the eyes, eyebrows etc.To find the landmarks on the faces, a shape estimator was used that is implemented in the dlib library based on the papers [12].After detecting the landmarks are converted into coordinates.

Along with landmarks we use head position and tilt to detect drowsiness. Head position is observed with respect to the calibrated camera's central axis.

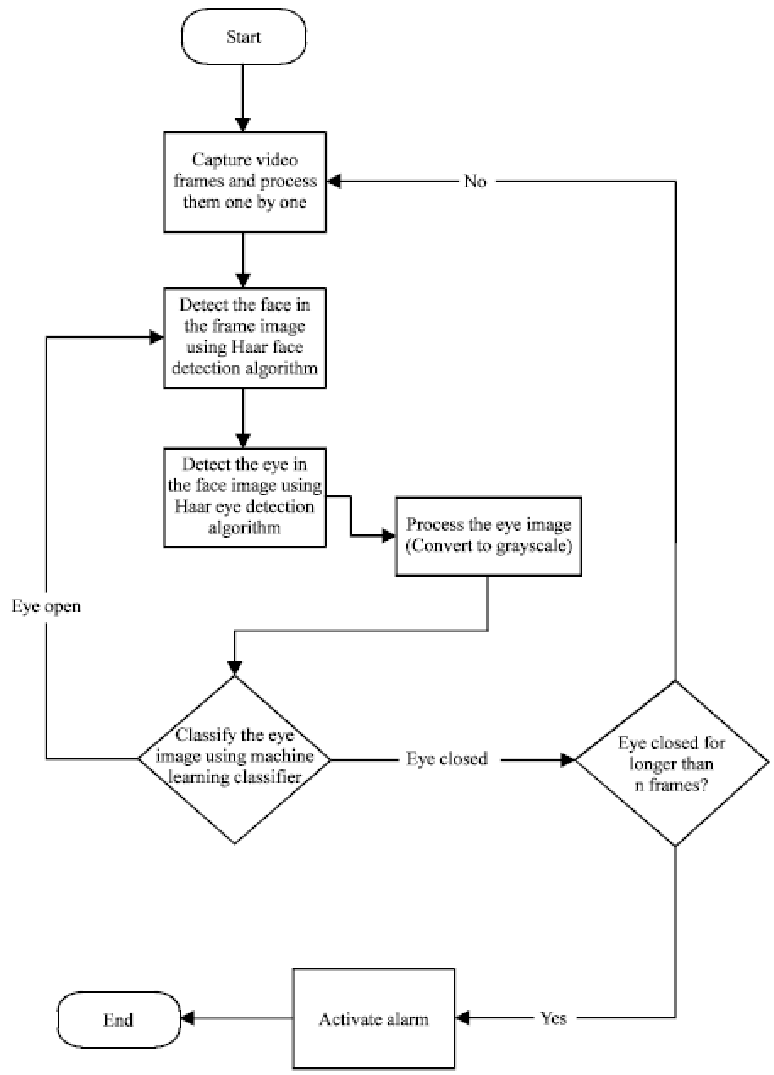

Figure 4.1: Flowchart Showcasing the methodology for Project [2]

Training of the data set using machine learning algorithm is the first step in voila Jones algorithm. Positive and Negative images are used for training.To develop predictive relationship between the data sets training is done . In Viola-Jones algorithm requires Haar-like features to be organized as classifier cascade. [8]The main advantage of a Haar-like feature is its calculation speed. AdaBoost Algorithm [5] is used for training the data set and selecting the required features.

AdaBoost finds a few good features which have significant variety. Dataset is trained after detemining the features.

4.1.2 Detection algorithm

4.1.3 Input

Real-Time Video Feed from Web Cam is currently being given as input to the application having Detection Algorithm (described below) to identify Facial Region.

4.1.4 Face detection

The face is detected using Viola Jones based Haar Classifiers [8]. The classifier is trained for a set of negative and positive images. The training algorithm produces the region of interest, here present in the green rectangular box; the negative images are the images that do not comprise faces and images that contain faces as positive samples of images. A sample of pixel values is generated by an input of face images. OpenCV thus acts as the trainer and the associated pixel values characterize an input feature as a non-face or face. If identified as a face, it is carried forward to detect the eyes and mouth, otherwise it is discarded. This identification is done with the help of cascading a number of classifiers .The advantage of using this is that rather than the entire set of features extracted together, it reduces computational load as each stage need only a certain set of the features to be extracted. [7]

4.1.5 Eye detection

Once the face is detected a region of interest (ROI) is selected from the face region, based on face geometry. Now in this ROI Haar Facial Landmarks are applied to detect eyes.Contours are drawn around the eyes.If it detects a closed eye a counter increments and camera fetches next frame and it is processed.In every frame the EAR value is calculated. From the threshold of EAR, fatigue level decision is taken, and can be used to alarm the driver.[7]

4.1.6 EAR (Eye Aspect Ratio)

Our algorithm determines the landmark positions , extracts a single scalar quantity EAR which defines opening of eye in frame.Ienerally not true that low value of the EAR means that a person is blinking. A low value of the EAR may occur when a subject performs a facial expression, yawning, etc., or the EAR captures a short random fluctuation of the landmarks. We have to find the (x, y) coordinates of facial landmarks within green bounding box. To determine these facial landmarks many algorithms have been developed. This algorithm can be divided into three categories; they are The Holistic Method, Constrained Local Method, and Regression-Based Methods. The performance of the holistic method and the Constrained Local Method is poor relatively to the Regression Based Methods [11]. Among the different types of Regression Methods, we used the Cascaded Regression Methods [11] to get our desired facial landmarks.

4.1.7 Blink detection

Once the eyes are detected next stage is the blink detection.We use a threshold value two determine whether the closed eye is a blink or part of the drowsy state.

68 facial landmarks are available on our face. But we just need the portion of the two eyes. So we cropped the portion of the two eyes along with facial

Figure 4.2: Cropped eye image with landmarks

landmarks which is shown in Fig. 2.3. Now from the landmarks of the eyes, the eye landmarks distance is calculated.

In Fig. 4.3, the eye landmarks as A, B, C, D, E, F are indicated. Then, the horizontal distances between A, E and B, D are calculated. Then we calculated the vertical distance between F, C. The distances between these landmarks are calculated from

$$D_{A,E} = \sqrt{(A_x - E_x)^2 + (A_y - E_y)^2}$$

$$D_{B,D} = \sqrt{(B_x - D_x)^2 + (B_y - D_y)^2}$$

$$D_{A,E} = \sqrt{(F_x - C_x)^2 + (F_y - C_y)^2}$$

4.1.8 MAR (Mouth Aspect Ratio)

MAR is defined simply as the relationship of the points shown below

It is evident that when the mouth is close the mouth aspect ratio almost zero. When the mouth is slightly open the mouth aspect ratio increases slightly. But where the mouth aspect ratio is significantly high it is clear that the mouth is wide open most probably for yawning.

$$\mathrm{MAR} = \frac{|CD| + |EF| + |GH|}{3 * |AB|}$$

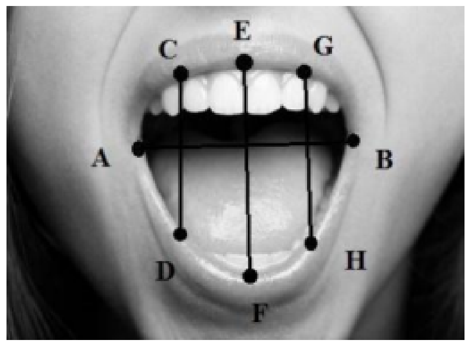

Figure 4.3: MAR

4.1.9 Head Tilt

The relative orientation and position with respect to a camera is referred to as pose of an object computer. In computer vision domain, The pose estimation problem is mostly referred as Perspective-n-Point problem or PNP. In this method our main aim is to find the pose of an face, provided we have calibrated camera, and we know the locations of n 3D points on the object and their 2D projections in the face.

It is possible to estimate the 3D rotation and translation of a 3D object from a single 2D photo, if an approximate 3D model of the object is known and the corresponding points in the 2D image are known. If we knew the translation and rotation of the pose, we could change the 3D points in world coordinates to 3D points in camera coordinates. The main purpose of head pose estimation is to calculate the rotation vector of the face in the current image. Here, them rotation vector corresponds to nodding(X axis), shaking(Y axis), and tilting (Z axis). The 3D points in camera coordinates can be projected onto the image coordination system using the intrinsic parameters of the camera .

In OpenCV the function solvePnP and solvePnPRansac can be used to estimate pose.

solvePnP implements several algorithms for pose estimation which can be selected using the parameter flag. By default it uses the flag SOLVEPNP-ITERATIVE which is essentially the Direct Linear Transform (DLT) solution followed by Levenberg-Marquardt optimization[15]. SOLVEPNP-P3P uses only 3 points for calculating the pose and it should be used only when using solvePnPRansac.

4.1.10 Detection System

Once the EAR, MAR values of the user are calculated for eyes and mouth respectively next step is to find the pose of the head i.e. its relative orientation and position with respect to the camera and then use all these factors to detect whether the person in a drowsy state or not. Basic principle is that the eyes are closed for longer durations when the person is drowsy than when the person is in active state and similarly for mouth , if it is open(¿ than threshold) and for longer duration then person is yawning.To check whether driver is paying attention to the road, we can use head pose estimation by using the camera looking at driver's face. If the eye aspect ratio exceeds the threshold then we can say that person is in a drowsy state. Similar conclusion can be said while calculating mouth aspect ratio.[7] It is known when human gets drowsy, the blood is moving into the end of hands and feet, and the eyes are blinked more often because tear production in the lachrymal glands is reduced. Hence Blink Rate is also used to effectively determine whether a person is drowsy or not.

Expected Output

The Detection Algorithm, if it finds the subject/driver in Drowsy State then an precautionary alarm would be sounded to bring driver's attention back. Also Real-Time Video Feed will have Facial Region highlighted in boxes and contours around eye region and mouth region along with a line from nose for head pose giving the head pose at an instant.

4.2 Work Done

The system proposed is built on Linux Operating system and the detection mechanism is carried out with the help of OpenCV Library. Linux is an interface between computer/server hardware, and the programs which run on it.Open source Computer Vision Library(OpenCV) is a library of programming functions that is exclusively used for applications based on computer vision.Attempts to detect drowsiness using OpenCV [5] has been carried out mostly under normal illumination before It has C++, C and Java interfaces and supports Linux easily. OpenCV[6] is quick when it comes to speed of execution.The core basis for Haar classifier object detection is the Haar-like features. These features use the change in contrast values between adjacent rectangular groups of pixels instead of the intensity values of a pixel. Features on the face are found using Facial Landmarks , generating 68 landmark points on the face .Ear and MAR are easily calculated using Euclidean distances between respective points of eyes and mouth.Head Pose is estimated using Perspective-n-Point method to find orientation of head relative to the camera.For calculating the 3D pose of an object in an image we need information about 2D coordinates of a few points, 3D locations of the same points and Intrinsic parameters of the camera.

4.2.1 Viola Jones Algorithm Using Haar classifier in OpenCV

Output of Viola Jones Algorithm is a trained Haar Classifier which is stored in XML File .All this has been done with the OpenCV training, which creates a classifier from a training set of samples with both postive as well as negative.

The work with a cascade classifier includes two major stages: training and detection. There are two applications in OpenCV to train cascade classifier: opencv-haartraining and opencv-traincascade. opencv-traincascade is a newer version, written in C++ in accordance to OpenCV 2.x API. But the main difference between this two applications is that opencv-traincascade supports both Haar and LBP(Local Binary Patterns) features. When talking about the quality of the two methods, LBP and Haar detection, it mostly depends upon how the training is done training includes the quality of the dataset and the training parameters associated with it.

Creating Samples

To prepare a dataset of positive samples on which training is to be done ,and test samples for testing opencv-createsamples function is used. The dataset of positive samples created by opencv-createsamples is in a format that is supported by opencv-haartraining and opencv-traincascade applications. The output that we get is a file with *.vec extension, which is in binary format containing images.

For training we need a set of samples. There are two types of samples: negative and positive.Positive images contains the object whereas negative images does not contain the object. Set of negative samples is made manually,opencv-createsamples utility creates a set of positive samples.

Negative Images

Negative images are those images which does not contain the object. These images should not contain the object that we are trying to detect (face). Negative samples are enumerated in a special file. it contains the image file name (directory of the description file) of negative images.Note that negative samples and sample images are also called background samples images or background samples. One index file, with list of image filenames is created (one per line)and these images are put into directories.

Positive Images

Positive images defines what the model should actually look for when trying to find your objects of interest(faces) by the boosting process. It supports two ways of generating a positive sample dataset:

1) We can generate a lot of positive images from a single positive image.

2) We can insert the positive images using an annotation tool to crop and resize the images and putting them in opencv format(binary).

Although first case works well for fixed objects(logo or a sign), it fails for objects that are less rigid (like faces).Second approach should be used in that case .As we want a flexible model, we take samples that covers a wide variety that can be in our class.In the case of faces you should consider different race groups, face complexion , beard styles and faces . Same applies for second approach.

Annotation Tool

For generating the info file, OpenCv community has provided with an annotation tool. The tool can be used by the command opencv-annotation if the OpenCV applications where build.

Example of usage: opencv-annotation - annotations=/path/to/annotations/file.txt - images=/path/to/image/folder/

This command will fire up a window containing the first image and your mouse cursor which will be used for annotation.Basically there are several keystrokes that trigger an action. The left mouse button is used to select the first corner of your object, then keeps drawing until you are fine, and stops when a second left mouse button click is registered. Finally you will end up with a usable annotation file that can be passed to the -info argument of opencv-createsamples.

4.2.2 Dataset information

Labeled Faces in the Wild, a database of face photographs designed for studying the problem of unconstrained face recognition. The data set contains more than 13,000 images of faces collected from the web. Each face has been labeled with the name of the person pictured. 1680 of the people pictured have two or more distinct photos in the data set. The only constraint on these faces is that they were detected by the Viola-Jones face detector. It was developed by Erik Learned-Miller, Gary B. Huang, Aruni Roy Chowdhury, Haoxiang Li, and Gang Hua.for University Of Massachusetts

4.2.3 Training

After the datasets is created we can now train classifier.Funtions and commands for training in OpenCv documentation.

As opencv-traincascade finishes , our trained new cascade is kept in cascade.xml file in the folder, passed as the parameter in the command.all the different cascade files are created for the case of interrupted training.

Parameters used- minHitRate= 0.98, Number of positive images= 1800, Number of negative images=900, Pixel width= 50px, Pixel height= 50px.

Training the cascade classifier took 4 days, 12 hours, 16 mins and 50 seconds.

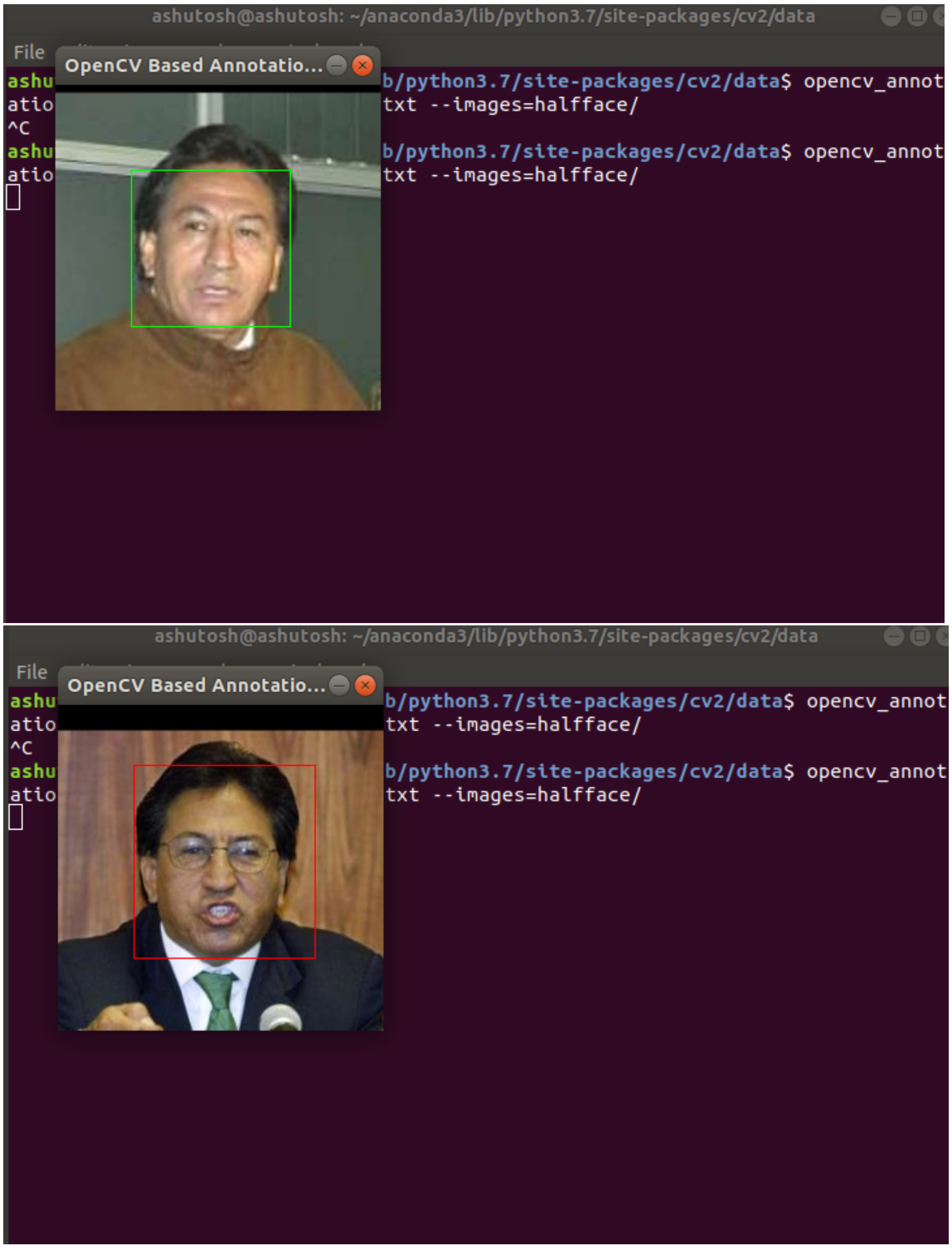

Figure 4.4: opencv-annotation command being used to select and confirm ROI in image

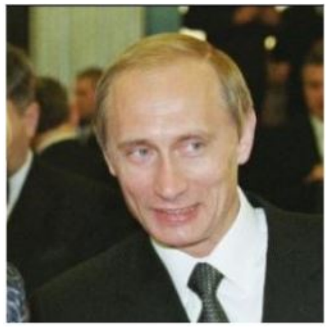

Figure 4.5: One of the images from labeled faces in the wild(lfw)

4.2.4 Facial Landmark

Localization and labelling of the areas like Mouth region, Right eyebrow region, Left eyebrow region, Right eye region, Left eye region and the Jaw region are detected by facial Landmarks.

After detecting the face and eye our further progress is dependent on the different facial features like the corner of the eyes. That's why facial landmarks detection is now necessary. Till now we have a bounding box around the face.

We have to find the (x, y) coordinates of the facial landmarks within the bounding box we achieved using viola jones algorithm. To find these facial landmarks many algorithms is developed till now. This algorithm can be divided into three categories; they are The Holistic Method, Constrained Local Method and Regression-Based Methods. The performance of the holistic method and the Constrained Local Method is poor in comparison with the Regression Based Methods. Among the different types of Regression Methods, we used the Cascaded Regression Methods [12] to obtain our desired facial landmarks

Mean eye landmarks distance calculation, now after calculating between the horizontal and vertical distances, the eye landmarks distances are calculated. Then, the eye landmarks distances for both eyes are calculated by

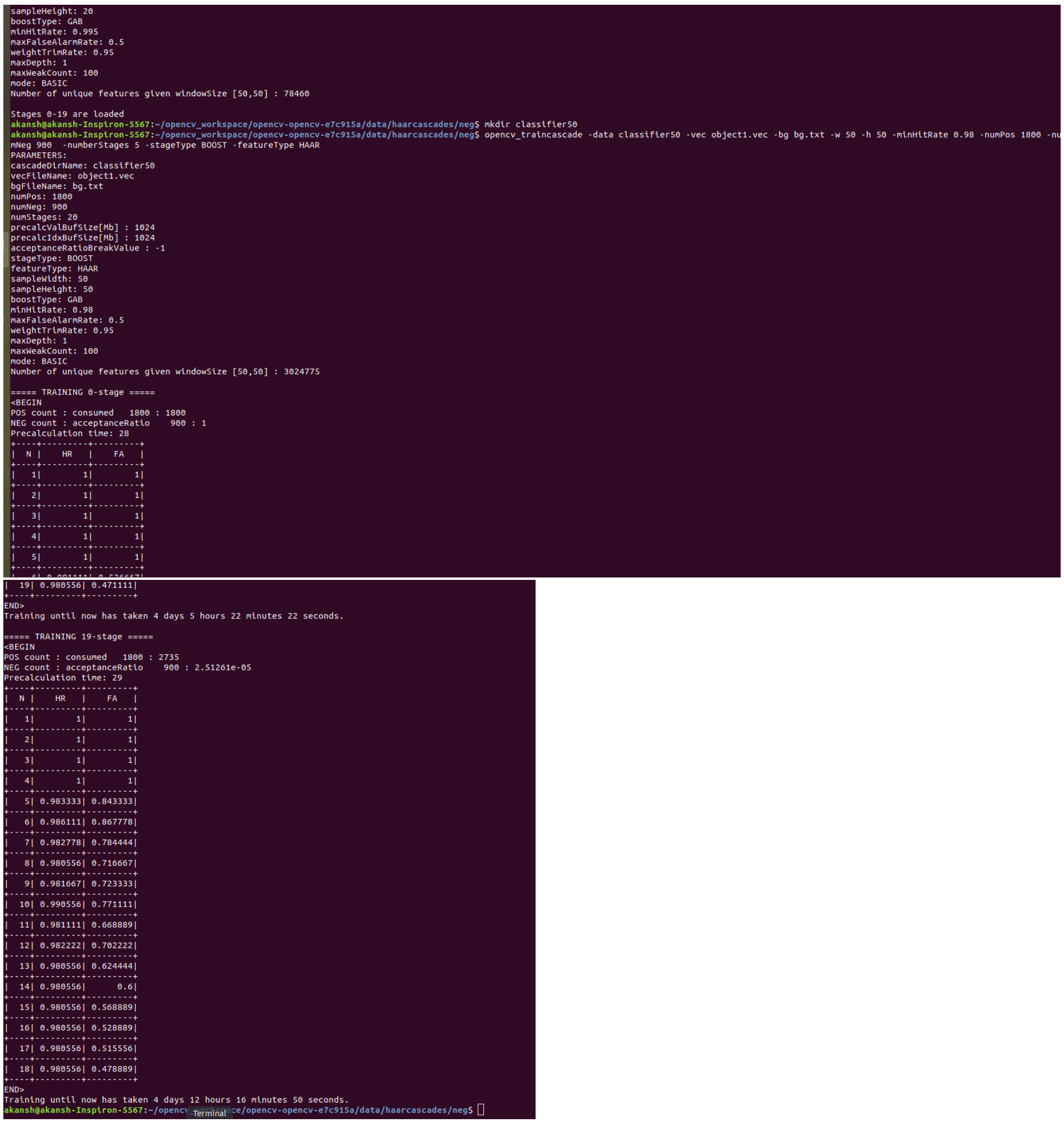

Figure 4.6: Training time

using equation mentioned above and then the average EAR is calculated Here, L is representing the eye landmarks distance. Now the eye landmarks distance of the left and right eye is calculated. For the left eye, the distance is l.l and l.r for the right eye. From these data, mean eye landmarks distance, L is calculated from

$$\bar{L} = \frac{l.r + l.l}{2}$$

Similar calculations have been done for MAR as mentioned in the formula provided before after which the value of MAR is also checked against a threshold to determine whether a person is yawning.When the mouth aspect ratio is significantly high it is clear that the mouth is wide open most probably for yawning.

4.2.5 Head Tilt estimation

Estimating the pose of 3D object is finding 6 numbers three for translation and three for rotation. The 3D coordinates of the various facial features shown above are in world coordinates. If we knew the rotation and translation (i.e. pose), we could transform the 3D points in world coordinates to 3D points in camera coordinates. The 3D points in camera coordinates can be projected onto the image plane (i.e. image coordinate system) using the intrinsic parameters of the camera (focal length, optical center etc.). In OpenCV the function solvePnP and solvePnPRansac can be used to estimate pose.Here, we are using solvePnP function. solvePnP function estimates the rotation vector and translation vector on the3D. Parameters used by the function:

objectPoints - Array of object points in the world coordinate space. I usually pass vector of N 3D points.

imagePoints Array of corresponding image points. we pass a vector of

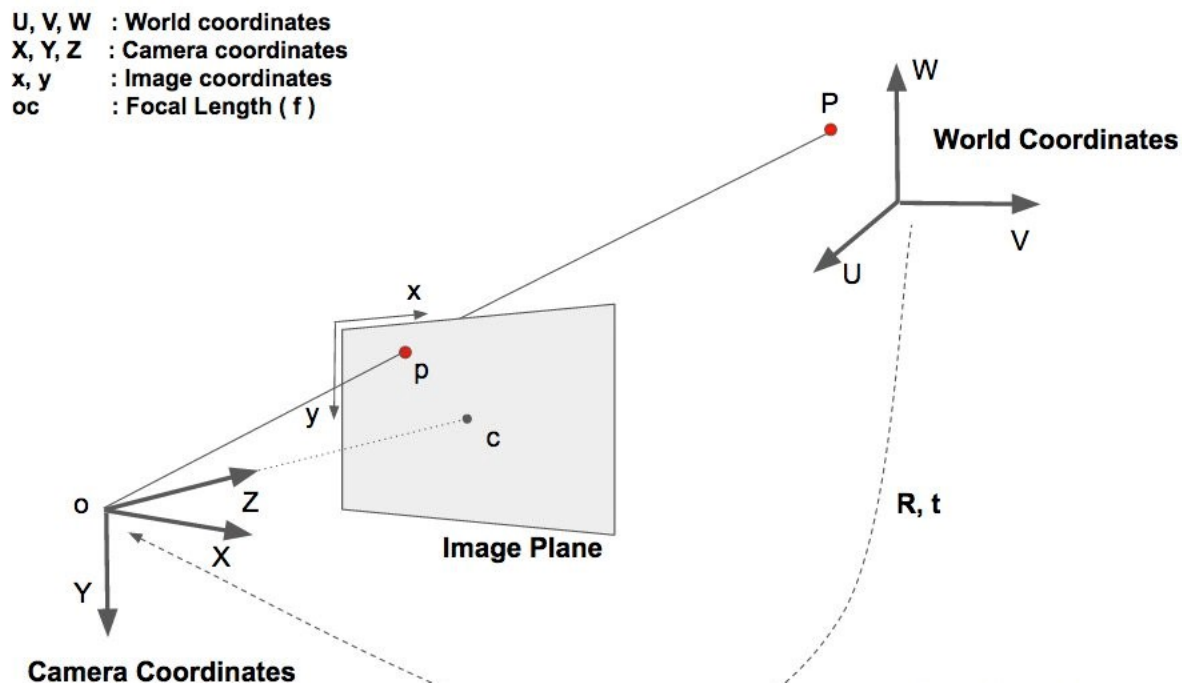

Figure 4.7: Projected 3D points onto the image plane

N 2D points found using the shape array(np).

cameraMatrix Input camera matrix A = $\begin{bmatrix} f-x & 0 & c-x \\ 0 & f-y & c-y \\ 0 & 0 & 1 \end{bmatrix}$

Camera Matrix has intrinsic parameters,which are the center points of image, (cx,cy); scale factor(s), and focal length between pixels (fx,fy).

distCoeffs Input vector of distortion coefficients. If the vector is NULL/empty, the zero distortion coefficients are assumed.

4.2.6 Alarm Detector

It generally does not hold that low value of the EAR means that a person is blinking.A low value of the EAR may occur when a subject closes his/her eyes intentionally for a longer time or performs a facial expression, yawning, etc., or the EAR captures a short random fluctuation of the landmarks.

Similarly, high value of MAR does not ensure a person is yawning. Drowsiness is then detected by computing the aspect ratios of eye frames and mouth based on there facial landmarks. The threshold for eye and threshold for mouth such that if eye aspect ratio (EAR) is less than threshold or if Mouth aspect ratio (MAR) is greater than threshold over a specified period of frames then drowsiness alert must be triggered.

$$\text{Eyeclosed} = \begin{cases} True & \text{if } EAR \leq \theta \\ False, & \text{otherwise.} \end{cases}$$

$$\text{Yawn} = \begin{cases} True & \text{if } MAR \geq \theta_1 \\ False, & \text{otherwise.} \end{cases}$$

A threshold distance 0.3 is set to differentiate between an open and close eye and the minimum number of consecutive frames for which eye ratio is below threshold for the alarm to be triggered is set as 50 as found from research papers placing this value in [45-50]. With the current set of hardware configurations our implementation gives optimal result at value of 50.

It has been found when human gets drowsy, the blood is moving into the end of hands and feet, and the eyes are blinked more often because tear production in the lachrymal glands is reduced. Hence Blink Rate is calculated using facial landmarks in mostly the same way as EAR.

3D coordinates of the various facial features are in world coordinates. If we knew the rotation and translation (i.e. pose), we could transform the 3D points in world coordinates to 3D points in camera coordinates. we get these vectors from solvePnp function and use these to form the projected coordinates using the 'projectPoints' function.We draw a line between the two points (line extending from nose) that provides with the head pose at any instant.If the 3D point in world coordinate differs largely from 3D point in camera coordinates then it can be concluded that the driver's attention is not on the road i.e. his head is tilted sideways(looking away).

We in our project have used all the above mentioned methods to collectively decide if the the driver is drowsy or concentration lapse occurs.Though good independently these factors never provide a good estimate. combining these methods covers most of the scenarios in which a driver is not attentive or drowsy.

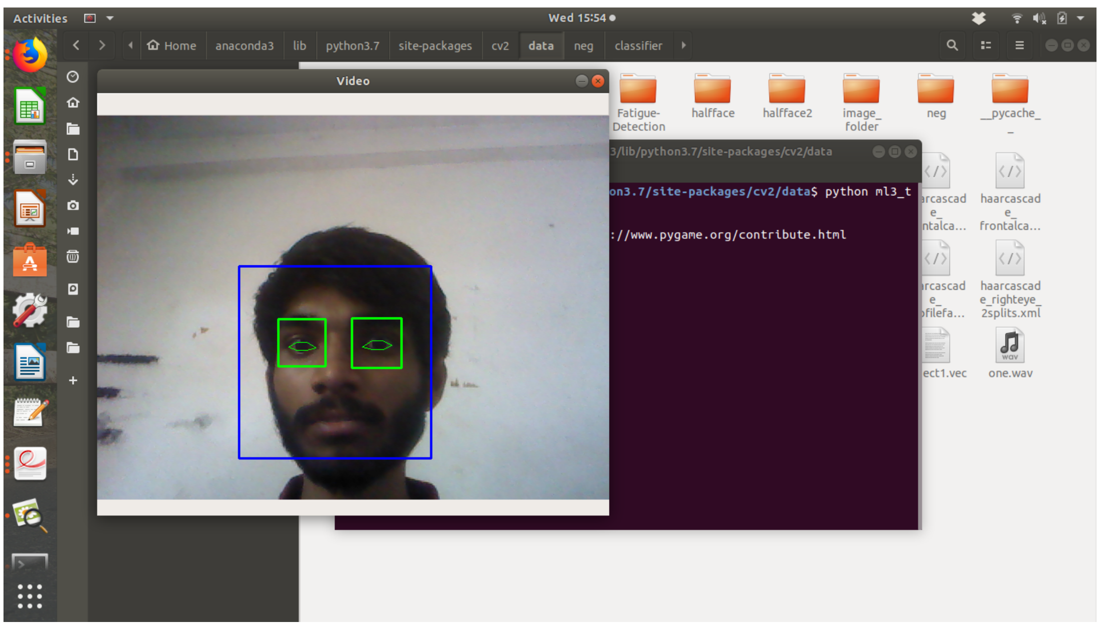

Figure 4.8: Eyes and face being detected and contours being drawn around eyes.

Chapter 5

Results

This project tries to combine all major features associated with drowsiness of a driver to achieve optimum result. The features being EAR, MAR, Blink Rate and Head tilt.

All the four parameters are considered simultaneously to determine whether a person is drowsy or not. Our trained haar cascade is better in identifying variations in facial structure, complexion along with identifying multiple faces in a single frame. Our haar cascade performs better in terms of computation time required to detect faces in real time scenario.

EAR and MAR values were observed for a specific time period in order

```
confusion_matrix:  [[103   6]
 [ 56 240]]
accuracy: 0.8469135802469135
Report :
              precision    recall  f1-score   support

           n       0.65      0.94      0.77       109
           y       0.98      0.81      0.89       296

   micro avg       0.85      0.85      0.85       405
   macro avg       0.81      0.88      0.83       405
weighted avg       0.89      0.85      0.85       405
```

Figure 5.1: Confusion matrix for the haar cascade tested on positive and negative images

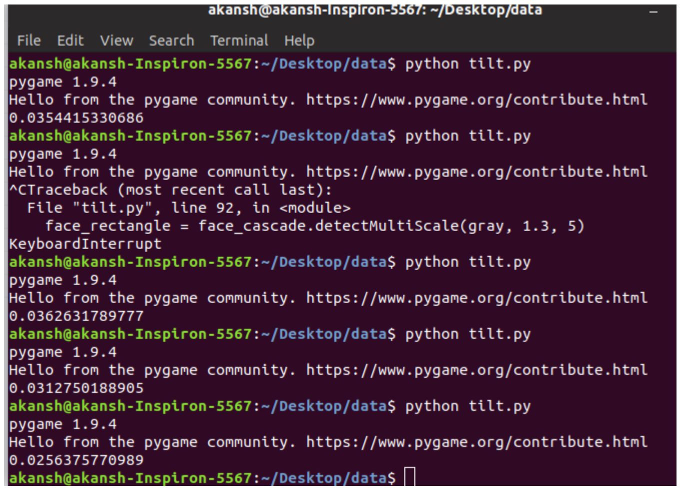

Figure 5.2: figure showing computation time of 'haar-cascade-frontalface-default.xml' and our haar cascade 'cascade.xml'

to find an approximate threshold value. Values obtained are as follows:

From the graph (Figure 5.3) it can be observed that the EAR value oscillates around 0.35 slightly when eyes are open but it drops drastically when the eyes are closed. It was observed that EAR value below 0.25 indicates that the person's eyes are closed.

From the graph (Figure 5.4) it can be observed that while a person is yawning MAR increases from 0.35 to 0.5-0.9 approximately. The threshold is observed as the lowest peak found while yawning.

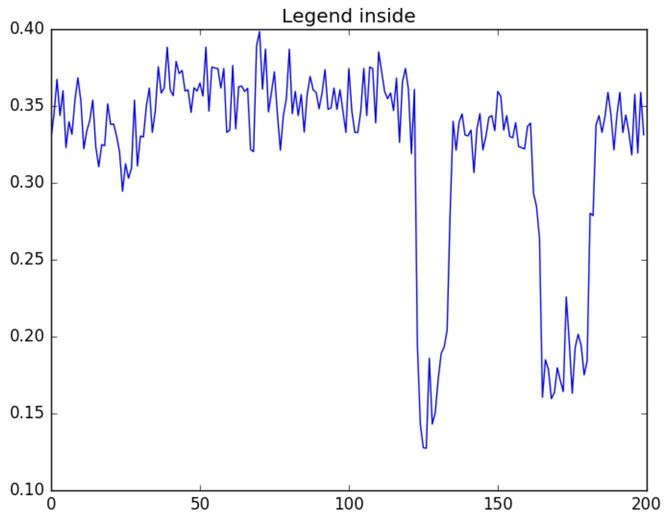

Figure 5.3: Graph of EAR against frames

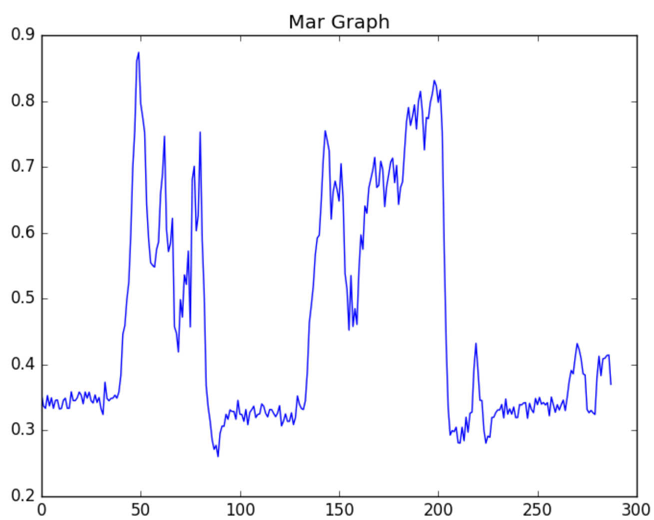

Figure 5.4: Graph of MAR against frames

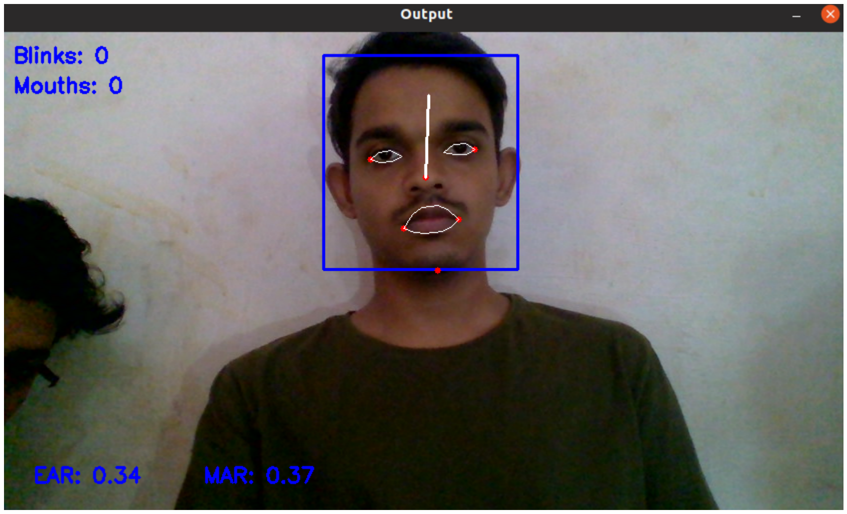

Figure 5.5: The image depicts face with landmarks marked and head pose being detected as it is evident that ear is above threshold and mar is below threshold hence no alarm(Normal Conditions)

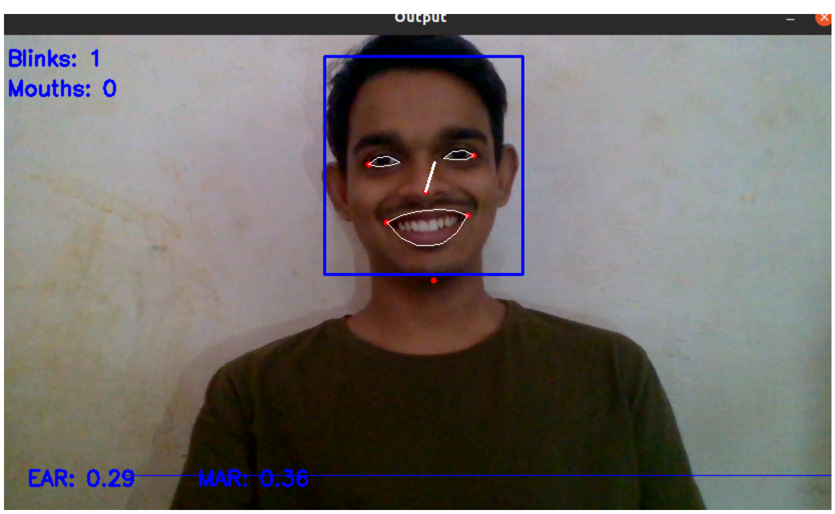

Figure 5.6: Though the person is smiling the mar values increases but not too much hence no alarm.

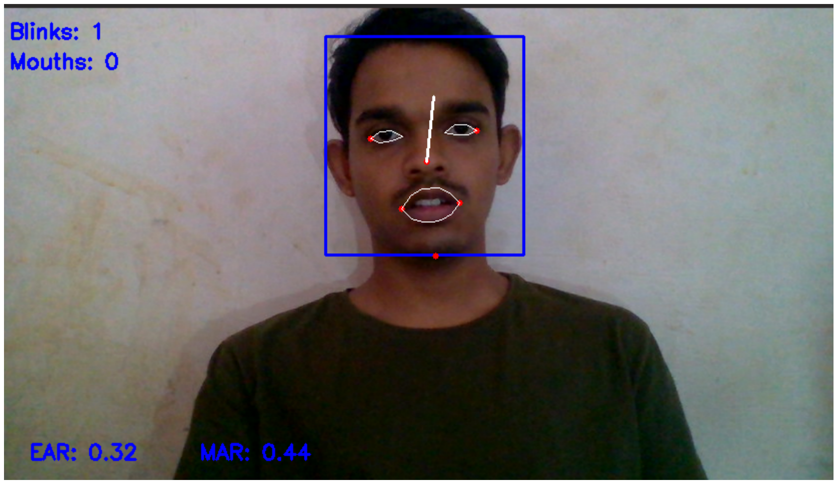

Figure 5.7: EAR remains above threshold and MAR remains below threshold while talking

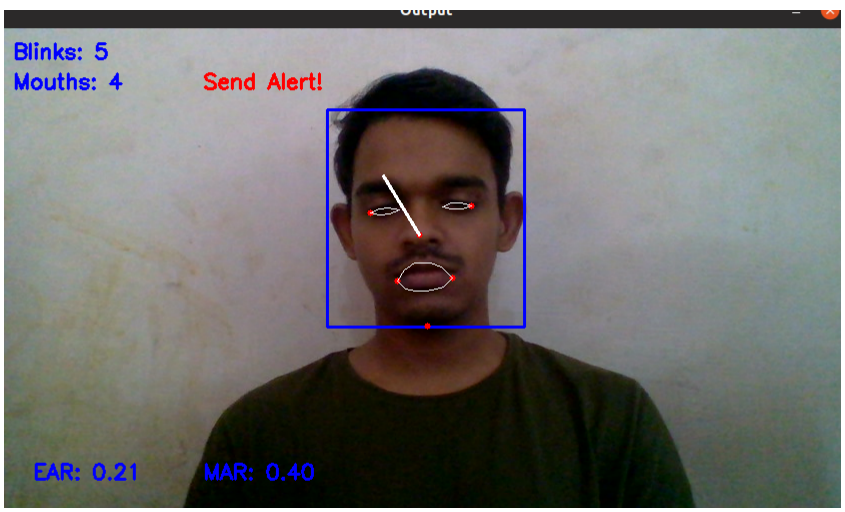

Figure 5.8: When the EAR goes below threshold, that is, person has his eyes closed for more than certain number of frames consecutively an alert is issued.

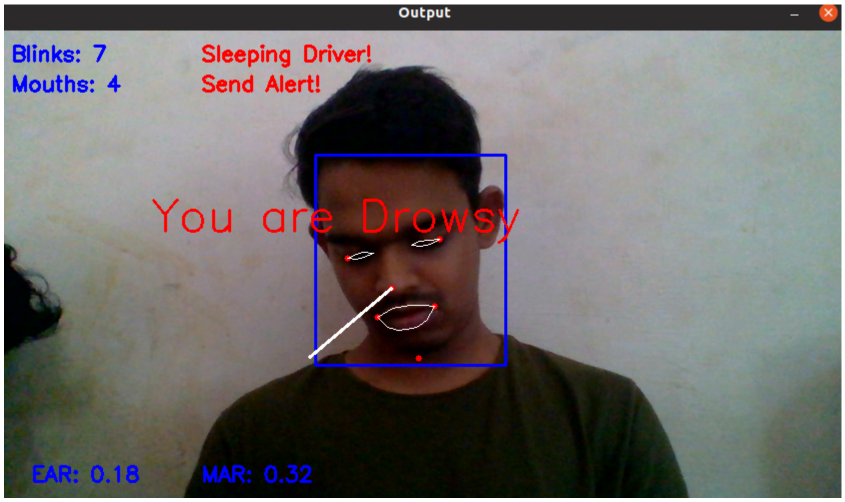

Figure 5.9: Drowsy alarm is sounded when EAR is below threshold and head pose is wrong too.

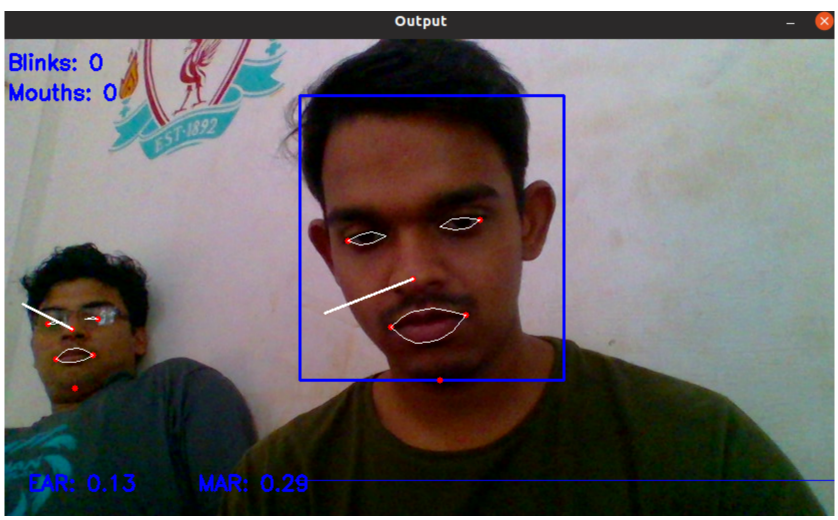

Figure 5.10: Detection of closest person to camera

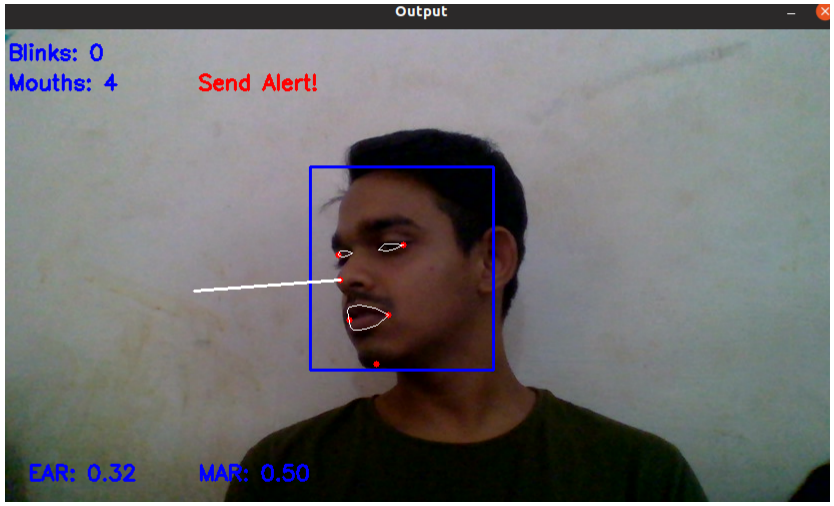

Figure 5.11: Alert while head tilt

Figure 5.12: Alarm sounded when person is drowsy, eyes closed and yawning

Chapter 6

Conclusion and Future work

6.1 Conclusion

The project intends to present a solution to alert the driver before a mishap happens. Detecting the driver drowsiness, which is one of the major cause of road accidents, will reduce deaths and injuries to a great extent. The Simulated system used to detect EAR, Eye blink, yawning and head pose estimation for drivers drowsiness/ attention using Viola Jones/haar cascade, facial landmark method and solvePnP (head tilt estimation). During monitoring the system is able to detect when the eyes are closed and mouth open simultaneously for too long and again and again in less period of time thus giving a buzzer sound to alert the driver. The system alerts the driver if he closes his eyes for long time which is giving information that the driver might have slept. Also the driver is alerted when his head is tilted or turned away using head pose estimation. The blinking of eye has been detected independent of haar classifiers are used for the eyes and Active contour method for yawning.

6.2 Future Work

In future, we will be trying to add real-time computation of threshold value to further improve our results. Also, we shall train our cascade on better hardware in order to improve accuracy in detecting faces.

References

[1] C. Murukesh, Preethi Padmanabhan, Department of Electronics and Instrumentation Engineering, Velammal Engineering College, Anna University, Chennai, INDIA. "DROWSINESS DETECTION FOR DRIVERS USING COMPUTER VISION",WSEAS Transactions on Information Science and Applications

[2] Ghassan Jasim AL-Anizy, Md. Jan Nordin and Mohammed M. Razooq,"AUTOMATIC DRIVER DROWSINESS DETECTION USING HAAR ALGORITHM AND SUPPORT VECTOR MACHINES TECHNIQUES"

[3] Mehul K Dabhi, Bhavna K Pancholi, M. S. University Baroda, India, " FACE DETECTION SYSTEM BASED ON VIOLA-JONES ALGORITHM", International Journal of Science and Research (IJSR).

[4] P. Viola and M. Jones, " RAPID OBJECT DETECTION USING BOOSTED CASCADE OF SIMPLE FEATURES" , Proceedings IEEE Conf. on Computer Vision and Pattern Recognition 2001.

[5] G. Bradski and A. Kaehler, "LEARNING OPENCV: COMPUTER VISION WITH THE OPENCV LIBRARY" , O'Reilly Media, Inc., 2008.

[6] Satya Mallick, "HEAD POSE ESTIMATION USING OPENCV AND DLIB — LEARN OPENCV", 2016

[7] Pauly Leo , Shankar Deepa ,Department Of Electronics and Engineering, CUSAT , "DETECTION OF DROWSINESS BASED ON HOG FEATURES AND SVM CLASSIFIERS" ,Proceedings Conference of IEEE International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN) 2015

[8] P. Viola and M. Jones, " ROBUST REAL-TIME FACE DETECTION" ,International Journal Of Computer Vision,Received September 10, 2001; Revised July 10, 2003; Accepted July 11, 2003

[9] Anjith George, "DESIGN AND IMPLEMENTATION OF REAL-TIME ALGORITHMS FOR EYE TRACKING AND PERCLOS MEASUREMENT FOR ON BOARD ESTIMATION OF ALERTNESS OF DRIVERS" , Report submitted to Indian Institute of Technology, Kharagpur , April 2012

[10] Tereza Soukupova , Jan Cech , "REAL-TIME EYE BLINK DETECTION USING FACIAL LANDMARKS" , Report submitted to 21st Computer Vision Winter Workshop Luka Cehovin, Rok Mandeljc, Vitomir Struc (eds.) Rimske Toplice, Slovenia, February 4, 2016

[11] Abdullah Arafat Miah , Mohiuddin Ahmad , and Khatuna Zannat Mim "DROWSINESS DETECTION USING EYE-BLINK PATTERN AND MEAN EYE LANDMARKS Distance ",International Joint Conference on Computational Intelligence (IJCCI 2018), At Dhaka, Bangladesh,December 2018.

[12] Nafis IRTIJA, Mahsius SAMI, Md Atiqur Rahman AHAD,"FATIGUE DETECTION USING FACIAL LANDMARKS",ISASE-MAICS 2018 .

[13] Punit Lohani, Rohan Putta ,Gayatri N Shinde,"REAL TIME DROWSINESS DETECTION SYSTEM USING VIOLA JONES ALGORITHM ",International Journal of Computer Applications (0975 8887) Volume 95 No.8, June 2014 .

[14] Vinay K Diddi1 , Prof S.B.Jamge, ”HEAD POSE AND EYE STATE MONITORING (HEM) FOR DRIVER DROWSINESS DETECTION: OVERVIEW”, Vol 1., Issue 9, Nov 2014

[15] Bilal Memon, Mansi Todi, Rubina Ngadong, ”DRIVER BEHAVIOR AND ALERT GENERATION USING HEAD MOVEMENT, EYE BLINKING AND YAWNING DETECTION”, Volume 5, Issue 06, June -2018