Sensors Based Automated System For Driver's Safety

Mr. Sunil Kumar S[1], Ms. Ramya S C[2], Ms. Sushma Shivananda Guttal[3], Mr. Sharath Babu T D[4], Mr. Vishwanath Muddi[5]

[1, 2, 3, 4] Dept of Electronics and Communication Engineering

[5]Asst. Professor, Dept of Electronics and Communication Engineering

[1, 2, 3, 4, 5] PESITM, Shivamogga, Karnataka, India

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 as 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 Open CV libraries are used including Haar-cascade. The entire system is implemented using Raspberry-Pi.

I. 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 driver drives for more than normal period for human then excessive fatigue is caused and also results in tiredness which drives the driver to sleepy condition or loss of consciousness.

Drowsiness is a complex phenomenon 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 includes Vehicle based measures, Physiological measures, Behavioral measures. Using those methods an intelligence system can be developed which would alert the driver in case drowsy condition and prevent accidents. The approach for entire system development is explained using a flow chart which includes capturing the image in real time continuously, then dividing it into frames. Then each frame is analyzed to find face first. If a face is detected then next task is to locate the eyes. After the positive result of detecting eye the amount of closure of eye is determined and compared with the reference values for the drowsy state eye. If drowsy condition is found out then driver is alarmed else repeatedly the loop of finding face and detecting drowsy condition is carried out.

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

In chapter 3, theoretical base for designing the entire system is explained which includes Principal Component Analysis (PCA) and Eigen face approach.

In chapter 4, the implementation part is described. The hardware that is used for the entire system is Raspberry Pi.

II. DROWSINESS DETECTION

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 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.

2. 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 occulogram)
EEG (electroencephalogram)

3. **Behavioral measures.**

Certain behavioral changes take place during drowsing like

1. Yawning
2. Amount of eye closure
3. Eye blinking
4. Head position

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.

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.

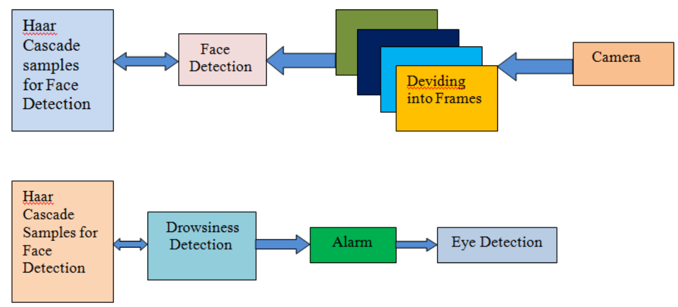

Figure 1.Flowchart of drowsiness detection system

Figure 2: Camera used for implementing 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 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.

III. 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.

IV. PRINCIPLE COMPONENT ANALYSIS

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.

4.1 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 x N matrix in N^2 dimension space. These N x 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 outs 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 liner 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.

4.1.1 Eigen Values and 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 scaler multiple of them. For the eigen vector X the obtained scaler called eigen value (λ). 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 $Det(A - \lambda I) = 0$, where 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 NxN then the solutions results in n roots of the characteristic polynomial. So it gives n Eigen values of A which satisfy the $Axi = \lambda ixi$, where i = 1,2,3,.....n. If the obtained eigen values are all distinct then we get n associated linearly independent eigen vectors with unique directions.

4.1.2 Face Image Representation

In this approach we represent set of let's say m images of each having size NxN. This is done by vectors of size N2. We represent each face Γ1, Γ2, Γ3… Γn. All those ɱobtained feature vectors are stored in the matrix with size NxN.

V. IMPLIMENTATION

The implementation of Fce detection is done with the help of

1. Raspberry
2. Webcam
3. Raspbian operating system.
4. Python IDLE.
5. OpenCv (Open source Computer Vision) for python with Haar 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.

5.1Viola-Jones algorithm has four stages:

1. Haar Feature Selection
2. Creating an Integral Image
3. Adaboost Training
4. Cascading Classifiers

5.1.1 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

The eyes region is darker than the upper-cheeks.

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.

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

5.1.2 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).

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 3. 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.

5.1.3 Ada Boost

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:

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

5.1.4 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.

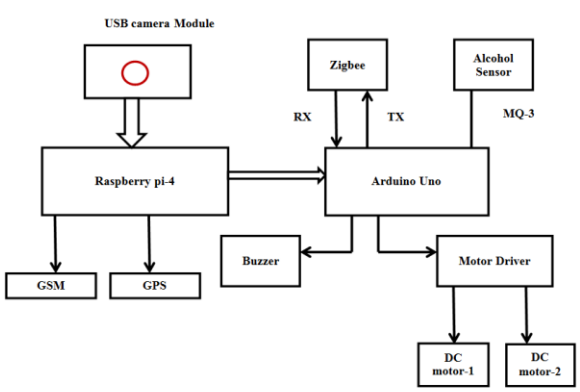

Figure 4. Block diagram of Drowsiness and Alcohol Detection.

VI. 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.

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.

VII. 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.

REFERENCES

[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.