Driver Drowsiness Detection System Using Raspberry Pi

D.ANAND[1], J.V.K.Vishwapoojitha[2], M.Sarumathy[3], R.Vaishaly[4]

[1]Assistant Professor, Department of Electronics and Communication Engineering,
[234]Final Year Students, Department of Electronics and Communication Engineering,
K.L.N. College of Engineering.
Email: anand.duraiswamy@gmail.com

Abstract—: A countless number of people drive on the highway day and night. People traveling long-distance suffer from lack of sleep. Due to which it becomes very dangerous to drive when feeling sleepy. The majority of accidents happen due to the drowsiness of the driver. According to a thorough investigation, there are over 500,000 accidents in India alone each year. Furthermore, driver fatigue is a factor in almost 60 percent of these accidents. In this paper, we provide a real-time monitoring system that makes use of face / eye and yarn identification and image processing algorithm. The role of image processing is to recognize the face of the driver and then extracts the image of the eyes of the driver for detection of drowsiness. The Haar face detection algorithm takes captured frames of image as input and then the detected face as output. It can be concluded this approach is a low cost and effective solution to reduce the number of accidents due to driver's Drowsiness to increase the transportation safety. Driver drowsiness detection is a vehicle safety technology which prevents accidents when the driver is getting drowsy.

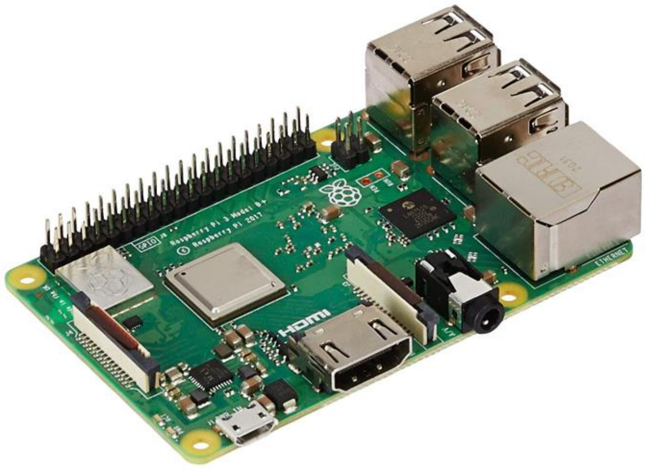
Keywords: Drowsiness, Face detection, Open Cv, Haar cascade

I. INTRODUCTION

Drowsy driving is one of the major causes behind fatal road accidents. One of the recent study shows that one out of five road accidents are caused by drowsy driving which is roughly around 21% of road accidents, and this percentage is increasing every year as per global status report on road safety 2015, based on the data from 180 different countries. This certainly highlights the fact that across the world the total numbers of road traffic deaths are very high due to driver's drowsiness. Driver fatigue, drink-and-drive and carelessness are coming forward as major reasons behind such road accidents. Many lives and families are getting affected due to this across various countries. Real time drowsy driving detection is one of the best possible major that can be implemented to assist drivers to make them aware of drowsy driving conditions. Such driver behavioral state detection system can help in catching the driver drowsy conditions early and can possibly avoid mishaps. With this paper, we are presenting technique to detect driver drowsiness using of Open CV, raspberry pi and image processing. Several studies have shown various possible techniques that can detect the driver drowsiness. Such driver drowsiness detection can be measured using physiological measures, ocular measure and performance measure. Among these physiological measure and ocular measure can give more accurate results. Physiological measure includes brain waves, heart rate, pulse rate measurements and these requires some sort of physical connection with the driver such as connecting electrode to the driver body. But this leads to discomfortable driving conditions. But ocular measure can be done without physical connection. Ocular measure to detect driver eye condition and possible vision based on eye closure is well suited for real world driving conditions, since it can detect the eyes open/ closed state using a camera. In Real Time Driver Drowsiness System using Image Processing, capturing drivers eye state using computer vision based drowsiness detection systems have been done by analyzing the interval of eye closure and developing an algorithm to detect the driver's drowsiness in advance and to warn the driver by in vehicles alarm. This section motivates how face is detected and how eye detection is performed for automotive application and their detection is necessary for assessing driver drowsiness.

Raspberry Pi 3:

The Raspberry Pi is a minimal expense, MasterCard estimated PC that plugs into a PC screen or TV, and utilizations a standard console and mouse. It is a fit little gadget that empowers individuals, everything being equal, to investigate figuring, and to figure out how to program in dialects like Scratch and Python.



Camera module:

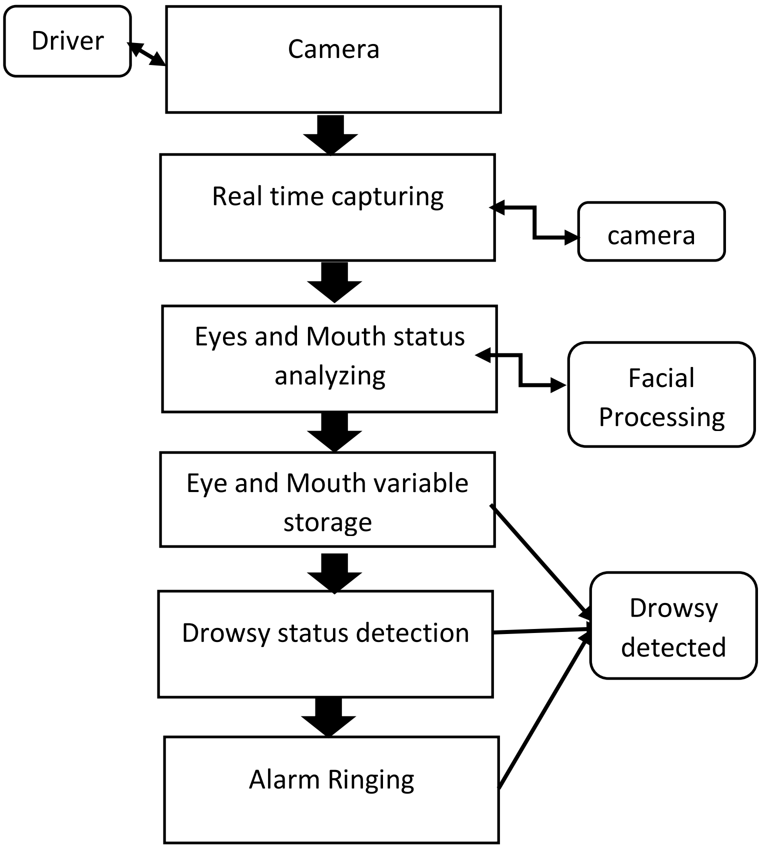
The flex link embeds into the connector marked CAMERA on the Raspberry Pi, which is situated between the Ethernet and HDMI ports. The link should be embedded with the silver contacts confronting the HDMI port. To open the connector, pull the tabs on the highest point of the connector upwards, then, at that point, towards the Ethernet port. The flex link ought to be embedded immovably into the connector, with care taken not to twist the flex at too intense a point. To close the connector, push the top piece of the connector towards the HDMI port and down, while holding the flex link set up

Buzzer:

There are two main types of buzzer: active and passive buzzer. A *passive* buzzer emits a tone when a voltage is applied across it. It also requires a specific signal to generate a variety of tones. The *active* buzzers are a lot simpler to use, so these are covered here. Connecting a buzzer an active buzzer connecting like an LED, But as they are a little more robust, you won't be needing a resistor to protect them.

II. SYSTEM ARCHITECTUR:

The architecture of the system consists of various phases



Camera:

This is the initial stage of the system. A setup is made and optimized for current user. Successful head location of the driver is the key stage. [4] If the driver's head is successfully located, it becomes easy to process the image and helps in identifying the current state of the driver.

Faces Detection:

The proposed system will start by capturing the video frames one by one. OpenCV provides extensive support for processing live videos. The system will detect the face in the frame image for each frame. This system uses Viola-Jones object detector which is a machine learning approach for visual object detection (Paul Viola, 2004 and Paul Viola, 2001). This is achieved by making use of the Haar algorithm for face detection. Haar cascade is a well-known robust feature-based algorithm that can detect the face image efficiently. With the use of uses of cascade of stages, Haar algorithm able to remove the candidates that are non -face. And each stage consists of combination of different Haar features and each feature in turn is classified by a Haar feature classifier. The inbuilt OpenCV xml “haarcascade_frontalface_alt2.xml” file is used to search and detect the face in individual frames. This file contains a number of features of the face and constructed by using a number of positive and negative samples. First load the cascade file then pass the acquired frame to an edge detection function, which detects all the possible objects of different sizes in the frame. Since the face of the driver occupies a large part of the image, instead of detecting objects of all possible sizes, specify the edge detector to detect only objects of a particular size i.e. for face region. Next, the output the edge detector is stored and this output is compared with the cascade file to identify the face in the frame. The output of this module is a frame with face detected in it. Only disadvantage in Haar algorithm is that it cannot extrapolate and does not work appropriately when the face is not in front of the camera axis. Once the face detection function has detected the face of the driver, the eyes detection function tries to detect the driver's eyes.

Eye Detection:

Once the face detection function has detected the face of the driver, the eyes detection function tries to detect the automobile driver's eyes. After face detection find eye region by considering eyes are present only in upper part of the face and from top edge of the face, extract eyes Region of Interest (ROI) by cropping mouth and hair, we mark it the region of interest. By considering the region of interest it is possible to reduce the amount of processing required and also speeds up the processing for getting exact eyes. After the region of interest is marked, the edge detection technique is applied only on the region of interest .Then search for eyes in ROI; Circular Hough Transformation is used here to find shape of eyes (Rhody Chester, 2005). The main advantage of the Hough transform technique is that it is liberal to gaps in feature boundary descriptions and is relatively unaffected by image noise, unlike edge detectors. The OpenCV function Hough Circles () is used to detect circles in an eye image. CHT ensure that at most two eyes found. With the eye detection technique we will only be able to detect the open state of eyes

Drowsy Detection:

After getting eyes the algorithm then counts the number of open eyes form each frame and determines the drowsiness. If the criteria are satisfied, then the driver is said to be drowsy. The buzzer connected to the system performs actions to correct the driver abnormal behavior. For this system, the eye and the face classifiers are required. The HARR Classifier Cascade files built-in there with the Open CV contains different classifiers for the face and eye detection."haarcascade_frontalface_alt2.xml"and function "Houghcircles ()" is used to search and detect the face followed by individual frames. The face detection and open eye detection have been carried out on each frame of the driver's captured facial image. The variable Eyes total is assigned to store the number of open eyes found in each frame. A variable will store the number of successive frames in which the eyes found to be closed with the values like 0, 1, 2, 3… etc. Initially, this variable is set to 0.When both the eyes are open, and then Drowsy count will be 0. Drowsy count will increase when Eyes total < 2. For an eye blink, Drowsy count value is raised by 1. If the eye blinks in more than 4 frames, i.e. variable count is greater than or equal to 4, then the condition for drowsiness is met and an alarm will be signaled at real time. For this system, the face and eye classifiers are required. The HARR Classifier Cascade files inbuilt on OpenCV include different classifiers for the face detection and the eyes detection. The inbuilt OpenCV xml "haarcascade_frontalface_alt2.xml" is used to search and detect the face in individual frames.

III. PROPOSED METHOD

The total system is separated into Training the dataset, Testing, and sending ready to alert the driver

Preparing

a) Initialize LBPH face recognizer.

b) Get appearances and IDs from the data set organizer to prepare the LBPH face recognizer.

c) Save the prepared information as a XML or YAML document. Load Haar classifier, LBPH face recognizer, and prepared information from XML or YAML document.

a) Capture the picture from the camera,

b) Convert it into grayscale,

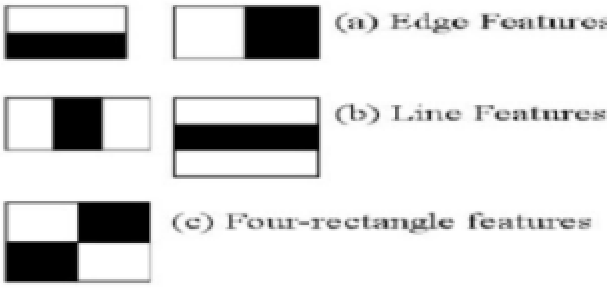
c) Detect the face in it and

d) Predict the face utilizing the above recognize

Raspberry Pi's control center either by utilizing SSH on a PC or by utilizing Keyboard and mouse with the showcase gadget like a TV associated with Pi. The calculation, first and foremost, needs a ton of positive pictures and negative pictures to prepare the Haar overflows classifier. Positive pictures are pictures with clear faces though bad pictures are those with no appearances.

Haar Cascades:

Each component is addressed as a solitary worth acquired from the distinction between the amounts of pixels in the white square shape from the amount of all pixels in the dark square shape. All various potential sizes and areas of the classifier are utilized for ascertaining a lot of elements. As the quantity of classifiers builds the number-crunching calculations appear to consume most of the day. To stay away from this, we utilize the idea of Integral Image. In Image Processing, an Integral picture is an information structure that is an added region table and calculation for rapidly and effectively creating an amount of values in a rectangular lattice subset. A basic picture is inferred by utilizing the equation.



Haar Cascade is based on the concept of features which are proposed by Paul Viola and Michael Jones in their paper "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It can be used to detect objects from an image or a video. This algorithm comprises of four stages:

a) Haar Feature Selection

b) Creating Integral Images

c) Adaboost Training

d) Cascading Classifiers

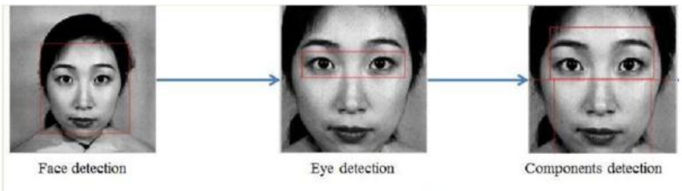


Fig. 3 Haar Cascade Feature Selection

IV. IMPLEMENTATION OF THE SYSTEM

A. Capturing the image

The camera module is set where individuals go into theeveryday schedule and video is taken inside a distance under 5 meters. A camera is utilized for taking video which contains different formats from which any of the housings can be utilized for confronting insistence and venturing the cooperation.

B. Detecting Faces:

Picking a successful calculation for face affirmation or acknowledgment is fundamental in this proposed work. There are various face location calculations available in OpenCV, for example, Eigenfaces, Fisher appearances, and Neighborhood Binary Pattern Histograms. Considering the expectations for the ongoing affirmation calculation which has been chosen is the Haar Cascade Algorithm for face recognition and affirmation. It is available in the OpenCV source library

C. Face Recognition :

The system enters into analysis stage after locating the driver's head, eyes and mouth properly in Video captured through camera. This video is preprocessed using various Image Processing techniques for drowsiness detection. Various techniques involved in Image Processing are Blurring, RGB to HSV Conversion, HSV Thresholding Blob Detection. In blurring stage, the previously captured video is decomposed into pixels which spreads out and gets mixed into surrounding pixels. This obtained image consists of some unique features which can be best expressed and described using HSV format. So the RGB image format is converted using HSV format. When the pixel color range is diverse, thresholding in HSV is very useful for isolating video features that cannot be achieved by RGB thresholding. So HSV thresholding is carried out.

After video thresholding blob detection method is implemented which aims at detecting regions of image that differ in properties like brightness or color as compared to surrounding regions. Driver's head movement is monitored using camera which is then analyzed using Centre of Gravity to detect the driver's state. For detecting if the driver is used which is placed right in front of the driver's face.

D. Alert Stage:

The system activates the alarm and alerts the driver if he/she is found in abnormal driving state i.e. being drowsy Then the alarm can be a buzzer (or) audio.