

# Driver Drowsiness Detection System Using Raspberry Pi

D.ANAND<sup>1</sup>, J.V.K.Vishwapoojitha<sup>2</sup>, M.Sarumathy<sup>3</sup>, R.Vaishaly<sup>4</sup>

<sup>1</sup>Assistant Professor, Department of Electronics and Communication Engineering,  
<sup>234</sup>Final Year Students, Department of Electronics and Communication Engineering,  
K.L.N. College of Engineering.

Email: [anand.duraiswamy@gmail.com](mailto:anand.duraiswamy@gmail.com)

**Abstract**— Day and night, the highway is filled with countless vehicles. Lack of sleep is a problem for long-distance travelers. Consequently, driving while sleepy is extremely risky. The cause of the majority of collisions is a driver who is fatigued. A thorough investigation revealed that over 500,000 accidents occur annually in India alone. Additionally, nearly 60% of these collisions are caused by driver fatigue. In this paper, we present a real-time monitoring system that uses an image processing algorithm and algorithms for face, eye, and yawn identification. The purpose of image processing is to identify the driver's face before extracting the driver's eye image to check for signs of intoxication. The captured image frames are used as input by the Haar face detection algorithm, which outputs the faces it has identified. **Conclusion:** This strategy is a low-cost and efficient way to increase transportation safety by reducing crashes caused by drowsy driving. A vehicle safety technology called driver drowsiness detection forgoes accidents by alerting the driver when they are about to fall asleep.

**Keywords:** Drowsiness, Face detection, OpenCv, Haarcascade.

## I. INTRODUCTION

One of the major factors contributing to fatal traffic accidents is drowsy driving. According to the 2015 Global Status Report on Road Safety, which is based on data from 180 different countries, one recent study indicates that one in five traffic accidents—or

roughly 21% of all accidents—are the result of drowsy driving. This percentage is also rising annually. This underlines the fact that drowsy driving contributes to a significant portion of all traffic fatalities worldwide. Road accidents are attributed to a number of factors, including driver fatigue, drunk driving, and carelessness. This is affecting a lot of people's lives and families in many different nations. One of the best strategies that can be used to help drivers become aware of drowsy driving conditions is real-time drowsy driving detection. A system like this one for detecting the behavioral state of the driver may be able to detect drowsiness in the driver early on and help prevent accidents. In this paper, we present a method for drowsy driving detection that makes use of Open CV, Raspberry Pi, and image processing. Studies have indicated a variety of potential methods for detecting driver intoxication. Utilizing physiological, ocular, and performance measurements, one can detect driver drowsiness. Ocular and physiological measurements among these can produce more precise results. The driver must be physically connected to the device in order to take physiological measurements such as heart rate, pulse rate, and brain waves. One way to do this is by attaching an electrode to the driver's body. However, this creates unpleasant driving conditions. However, ocular measurement can be performed without a physical link. Ocular measures that assess a driver's vision and eye health based on eye closure are well suited to the conditions of real-world driving because they can detect whether the eyes are open or closed using a camera. In a Real time Driver Drowsiness System using Image Processing, capturing the

driver's eye state using computer vision-based drowsiness detection systems has been done by analyzing the time between eye closure and developing an algorithm to detect the driver's drowsiness in advance and to warn the driver by in-vehicle alarm. This section explains why detecting faces and eyes is important for determining driver drowsiness in the context of automotive applications.

### Raspberry Pi 3:

The Raspberry Pi is a low-cost, under-\$100 PC that uses a standard console and mouse, plugs into a PC or TV screen, and is estimated by MasterCard. With all things being equal, it is a useful little device that enables people to learn how to program in languages like Scratch and Python and to explore mathematics.

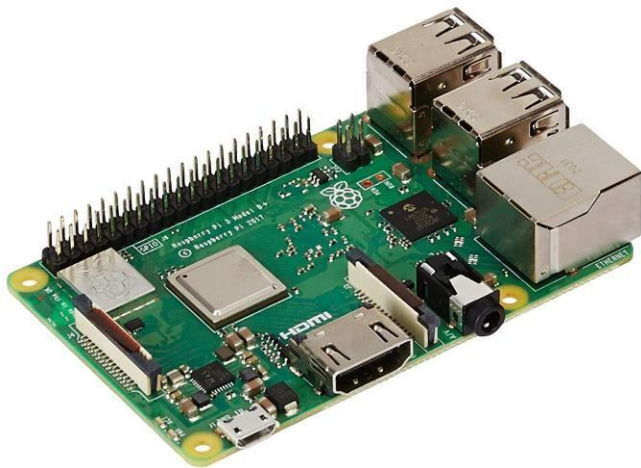


Fig: 1

### 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.



Fig: 1.1

### Buzzer:

Active and passive buzzers are the two main categories of buzzer. When a voltage is applied across a passive buzzer, a tone is released. The generation of various tones also needs a particular signal. These are covered because active buzzers are much easier to use. While you can connect an active buzzer like an LED, you won't need to use a resistor to protect them because they are a little more durable.



Fig: 1.2

## II. SYSTEM ARCHITECTURE:

The architecture of the system consists of various phases

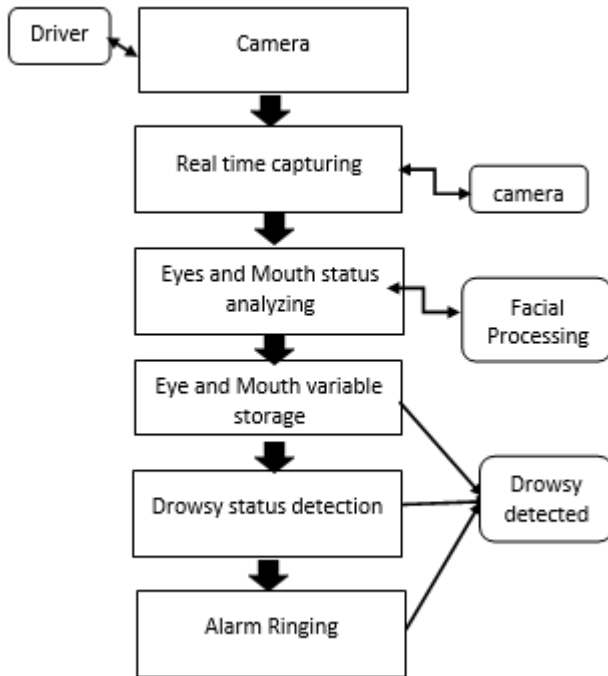


Fig: 2

### Camera:

The system is currently in its initial stages. For the current user, a setup is created and optimized. The crucial step is for the driver's head location to be successful. [4] If the driver's head is found, it is then simple to process the image and determine the driver's current condition.

### Face Detection:

The suggested system will begin by individually capturing each video frame. Processing live videos is greatly facilitated by OpenCV. Each frame's face will be recognized by the system. According to Paul Viola (2004, 2001), this system employs the Viola-Jones object detector, a machine learning method for detecting visual objects. The Haar face detection algorithm is used to achieve this. The well-known robust feature-based algorithm Haar cascade is effective at detecting faces in images. The Haar algorithm is able to eliminate candidates who lack facial features by employing a cascade of stages. Additionally, each stage consists of a combination of

unique Haar features, and a Haar feature classifier classifies each feature separately. "Haarcascade\_frontalface\_alt2" is an embedded OpenCV xml file. The face in each frame is searched for and found using an xml file. This file, which was created using both positive and negative samples, includes a number of facial features. The cascade file must first be loaded before the acquired frame can be passed to an edge detection function, which finds any potential objects of various sizes in the frame. Since the driver's face takes up a significant portion of the image, you should instruct the edge detector to only pick up objects that are that specific size i, rather than all possible sizes. E. Concerning the face. The edge detector's output is then saved and used to identify the face in the frame by comparing it to the cascade file. A frame with a face detected in it is the module's output. The only drawback of the Haar algorithm is that when the face is not in front of the camera axis, it cannot extrapolate and does not function properly. Once the driver's face has been identified by the face detection function, the eyes detection function tries to identify the driver's eyes.

### Eye Detection:

Following the driver's face being identified by the face detection function, the eyes detection function looks for the driver's eyes. Following face detection, locate the eye region by assuming that eyes are only present in the upper portion of the face, and then, starting from the top edge of the face, extract the eyes Region of Interest (ROI) by cropping the mouth and hair, then we mark it as the region of interest. It is possible to shorten the processing time and accelerate the process for getting precise eyes by taking the area of interest into account. Following the marking of the region of interest, the edge detection technique is only used on the ROI. Next, look for eyes in the ROI; the Circular Hough Transformation is used to determine the shape of the eyes in this instance (Rhody Chester, 2005). Contrary to edge detectors,

the Hough transform technique has the main advantage of being forgiving of gaps in feature boundary descriptions and being comparatively unaffected by image noise. To find circles in an image of an eye, use the OpenCV function Hough Circles (). CHT guarantees that there will only be two eyes. We will only be able to detect open eyes using the eye detection technique.

### **Drowsy Detection:**

After getting eyes the algorithm then counts the number of open eyes from 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's abnormal behavior. For this system, eye and face classifiers are required. The HARR Classifier Cascade files built-in there with the Open CV contains different classifiers for 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 the 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, face and eye classifiers are required. The HARR Classifier Cascade files inbuilt on OpenCV include different classifiers for 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.

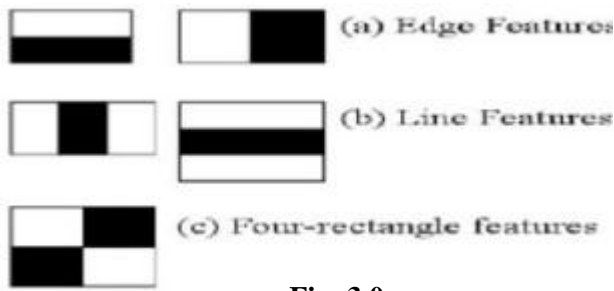
## **III. PROPOSED METHOD**

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.
- d) Capture the picture from the camera,
- e) Convert it into grayscale,
- f) Detect the face in it and
- g) 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:**

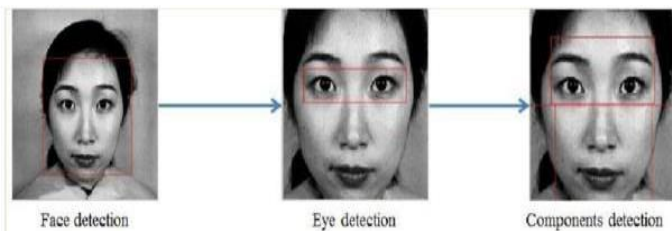
Each component is addressed as a solitary worth acquired from the distinction between the amounts of pixels in the white square shape from the amount of all pixels in the dark square shape. All various potential sizes and areas of the classifier are utilized for ascertaining a lot of elements. As the 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.



**Fig: 3.0**

The foundation of Haar Cascade is the idea of features that Paul Viola and Michael Jones put forth in their 2001 paper, "Rapid Object Detection using a Boosted Cascade of Simple Features.". Using machine learning, a cascade function is trained using a large number of both positive and negative images. It is possible to use it to find objects in a picture or a video. Four stages make up this algorithm.

- Haar Feature Selection
- Creating Integral Images
- Adaboost Training
- Cascading Classifiers



**Fig. 3 Haar Cascade Feature Selection**

## IV.IMPLEMENTATION OF THE SYSTEM

### A. Capturing the image

A video is recorded from a distance of no more than five meters using a camera module that is placed where people go about their daily activities. Any of the housings can be used from a camera to capture video in a variety of formats for use in defying resistance and attempting 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:

Once the driver's head, eyes, and mouth have been correctly located in the video that was captured by the camera, the system moves on to the analysis stage. Various image processing techniques were used to preprocess this video in order to detect drowsiness. Blurring, RGB to HSV Conversion, and HSV Thresholding Blob Detection are just a few of the many image processing techniques. During the blurring stage, the pixels from the previously recorded video are broken down, spread out, and mixed with nearby pixels. HSV format is the best way to express and describe the special characteristics of this obtained image. HSV format is used to convert RGB image format. Thresholding in HSV is incredibly helpful for isolating video features that can't be accomplished by RGB thresholding when the pixel color range is varied. HSV thresholding is thus performed.

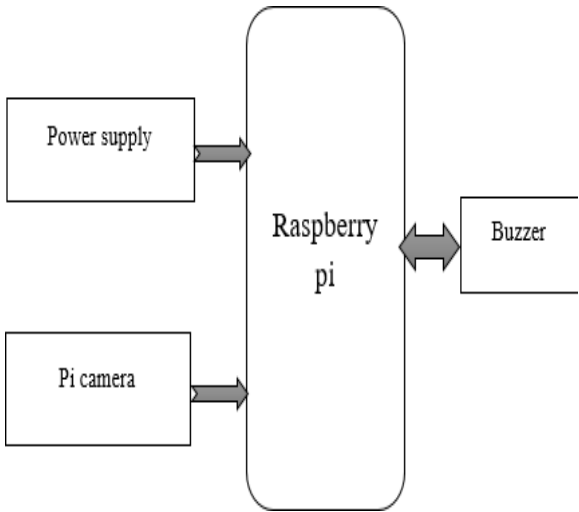
Blob detection is a technique used after video thresholding that seeks to identify areas of an image that have different brightness or color values from nearby areas. Camera is used to track the driver's head movements, and Center of Gravity is used to analyze the data to determine the driver's condition. It is placed directly in front of the driver's face to detect whether the driver is present.

### D. Alert Stage:

If the driver is discovered to be in an abnormal driving state, the system activates the alarm and notifies them. E. The alarm can then be an audio buzzer if you are sleepy.



**BLOCK DIAGRAM**



**Fig: 4**

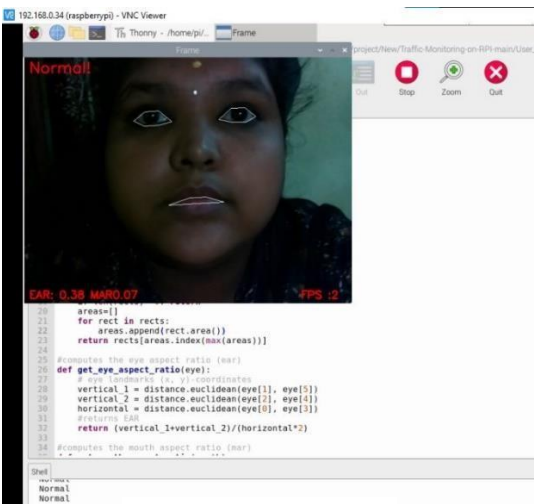
**Setup:**



**Fig: 4.1**

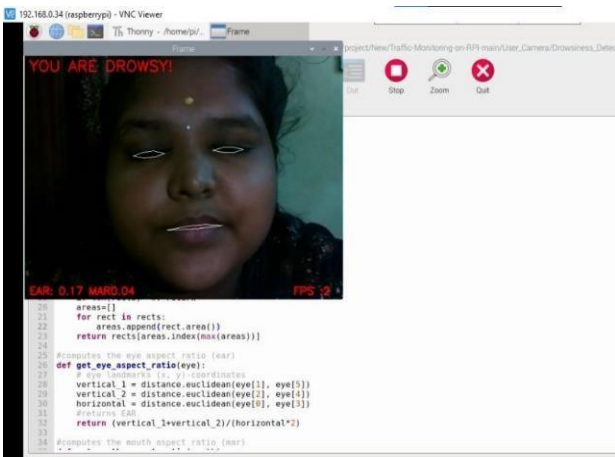
**Output:**

**A. Face recognition**



**Fig: 5**

**B. Eye detection**



**Fig: 5.1**

**C. Yawn detection**



**Fig: 5.2**

## CONCLUSION & FUTURE SCOPE:

A driver's drowsiness, intoxication, or reckless behavior can be quickly identified by a system developed to monitor driver abnormalities. The driver's eye closure served as the basis for the development of a drowsiness detection system that can distinguish between normal eye blinking and drowsiness while also detecting drowsiness while driving. Accidents caused by drivers who are sleepy can be avoided with the proposed system. The system functions well even when drivers are wearing eyeglasses and in low light situations if the camera produces better results. Several in-house image processing algorithms are used to gather information about the position of the head and eyes. The monitoring system can determine whether the eyes are open or closed. There is a warning signal that is given when the eyes are closed for too long. Continuous eye closures are used by processing to determine the driver's level of alertness.

Drowsiness detection systems may be added to aircraft in the future to warn pilots when a pilot is getting sleepy.

- Drunk drivers can also use the alcohol sensor.

- Drowsiness detection systems can be implemented in schools and colleges in the future to alert staff members when a student is showing signs of being sleepy in a class.

## REFERENCE

- [1]. Kouzani A, He F, Sammut K. Fractal face representation and recognition. IEEE Int Conf Syst Man Cybern 1997;2.<http://dx.doi.org/10.1109/ICSMC.1997.638231>
- [2]. Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on (Vol.1,pp.I-511).IEEE
- [3]. Bagus G. Pratama, IgiArdiyanto, Teguh B. Adji, A Review on Driver Drowsiness Based on Image, Bio-Signal, and Driver Behavior, IEEE, July 2017.
- [4]. E. Vural, M. Cetin, A. Ercil, G. Littlewort, M. Bartlett, and J. Movellan, "Drowsy driver detection through facial movement analysis," in International Workshop on Human-Computer Interaction, 2007, pp. 6- 18: Springer.
- [5]. P.Thiffault and J. Bergeron, "Monotony of road environment and driver fatigue: a simulator study," Accident Analysis & Prevention, vol. 35, no. 3, pp. 381-391, 2003.
- [6]. "Drowsy Driving," National Highway Traffic Safety Administration (NHTSA), 01-Feb-2018.[Online].
- [7].FACES ALGORITHM", ARPN Journal of Engineering and Applied SciencesVOL.11,NO.13
- [8].Rafael C. Gonzalez. 2009. Digital Image Processing. Pearson Education India.
- [9]. Kyong Hee Lee, Whui Kim, Hyun Kyun Choi, Byung Tae Jan. A Study on Feature Extraction Methods Used to Estimate a Drivers Level of Drowsiness, IEEE, February 2019.

[10]. Fouzia, Roopalakshmi R, Jayantkumar A Rathod, Ashwitha S, Supriya K, Driver Drowsiness Detection System Based on Visual Features. , IEEE, April 2018.