Driver Drowsiness Detection System using Machine Learning Algorithms

Shivani Sheth, Aditya Singhal, V.V. Ramalingam

Abstract: Road crashes are the most common forms of accidents and deaths worldwide, and the significant reasons for these accidents are usually drunken, drowsiness and reckless behaviour of the driver. According to the World Health Organization, road traffic injuries have risen to 1.25 billion worldwide, which makes driver drowsiness detection a major potential area to avert numerous sleep-induced road accidents. This project proposes an idea to detect drowsiness using machine learning algorithms, hence alarming the driver in real-time to prevent a collision. The model uses the Haar Cascade algorithm, along with the OpenCV library to monitor the real-time video of the driver and to detect the eyes of the driver. The system uses the Eye Aspect Ratio (EAR) concept to determine if the eyes are open or closed. We also feed a data-set file consisting of the facial features data-points to train the machine learning algorithm. The model inspects each frame of the video, which helps to recognize the state of the driver. Furthermore, a Raspberry Pi single-board computer, combined with a camera module and an alarm system, facilitates the project to emulate a compact drowsiness detection system suitable for different automobiles.

Keywords: Raspberry Pi, OpenCV, Haar Cascades, Image Processing, Eye Detection, Driver drowsiness, Alarm System, Real time

I. INTRODUCTION

The automobile industry in today's time has shown a steady rise across the globe. Consequently, the number of vehicles is increasing exponentially, which has further led to an increase in road accidents in each country. The road accidents have proved to be a menace that has majorly reduced the safety of the general public, let alone the driver. The World Health Organization identified sleepiness, alcoholism, and carelessness as the significant causes of road accidents in their Global Status Report on Road Safety. As a result, the fatalities and associated expenses that follow prove to be a severe threat to families across the world. The current drowsiness detection methods used are not widespread due to their high cost and less availability, thus making them unfeasible in the usual or non-luxury vehicles. Therefore there is a growing need for a smart and viable drowsiness detection system that the numerous automobiles in the industry can quickly adapt. The fields of machine learning and artificial intelligence have made numerous groundbreaking advances, which use different algorithms to train the model to be smart and autonomous.

Revised Manuscript Received on February 12, 2020.

Shivani, Sheth Computer Science and Engineering SRM Institute of Science and Technology Chennai, India shivani29111998@yahoo.com

Aditya, Singhal Computer Science and Engineering SRM Institute of Science and Technology Chennai, India aditya12singhal@gmail.com

V.V.Ramalingam, Computer Science and Engineering SRM Institute of Science and Technology Chennai, India ramalinv@srmist.edu.in

This model proposes one such algorithm, known as the Haar Cascade algorithm, which is combined with the various libraries in Python to capture and detect drowsiness in real-time. This model is efficient in detecting drowsiness as the algorithm is optimal in speed and accuracy when combined with a mini processing unit, such as a Raspberry Pi, as used in this project. This compact system is a suitable measure of detection in the automobiles and does not cause any hindrance or difficulties to the driver. Hence, the application of the given new tools and algorithms could constitute a powerful technology for the driver drowsiness detection system in real-time. It can also be widely used among different drivers and can be adapted by a variety of vehicles.

II. STATE OF THE ART

Numerous techniques have been implemented in the field of driver drowsiness detection, which includes various machine learning algorithms. In an attempt to increase the accuracy and accelerate the process of drowsiness detection, many methodologies have come up based on the different arenas of drowsiness detection as described below:

- Driving Patterns: This approach takes into consideration the trajectory or the path followed by a driver on the road. External factors such as the condition of the road, the type of vehicle and the environmental situations greatly influence the model. It calculates the deviation of the vehicle based on the position of the automobile at a specific time. This deviation calculates the driving patterns of the driver based on which the model determines if the driver is drowsy [1]. Thus, the model is not always accurate because of its dependency on various external conditions.
- Physiological Sensors: This type of drowsiness detection system depends on the Electroencephalogram (EEG), Electrocardiogram (ECG), and Electrocardiographs (EOG) physiological sensors that detect the state of the human brain. It determines if a person is drowsy by sensing the electric currents of the body and the brain [2]. Although the system has an accuracy rate of above 90%, it proves to be unfeasible for drowsiness detection in automobiles due to the number of sensors that have to be attached to the human body, which could cause discomfort to the driver and hinder his/her driving.
- Computer Vision: These drowsiness detection systems use image processing to detect the facial features, such as the eyes and the mouth of the driver, using different machine learning algorithms. They are deployed on a computer capable of processing the algorithm and is more suitable for drowsiness detection in automobiles.

It uses various detection methods to determine drowsiness, such as eye detection or yawn detection, to recognize if the driver is drowsy.

The most feasible method for a Driver Drowsiness Detection System among the three, as seen above, is the Computer Vision method. This is because it neither relies on any external factor that could lead to a false positive nor does it require any physical connections to the driver that could distract the driver. The computer vision domain uses a variety of machine learning algorithms to determine drowsiness, such as the Support Vector Machine (SVM) algorithm that classifies objects by separating data items [3]. It detects the eyes and other facial features using a dataset but gives less accurate results and has a higher error rate, especially in large or noisy datasets. Another such algorithm is the Convolutional Neural Networks (CNN) model, which performs drowsiness detection using neural networks that imitate the working on the human brain on a computer [4, 5]. It proves to be considerably accurate but also requires a high computational cost and a large dataset to train the model due to which it is not the best fit for our drowsiness detection system. The last significant model that we consider for our project uses the Haar Cascades algorithm, which uses the facial features of the driver to detect drowsiness [6]. It is the second most accurate and fastest algorithm after the CNNs, and also works on low computational cost and a smaller training set, which makes the system economical and the most suitable model for our purpose.

III. PROPOSED WORK

The proposed system architecture mainly consists of five modules, namely the Camera module, Raspberry Pi, the Python libraries, the Display, and the Buzzer:

- Camera Module: The camera module provides a continuous real-time video of the driver from which the model extracts each frame used to detect the state of the eyes. It presents as an input to our proposed model.
- Buzzer: The buzzer is an output device, which is connected to Raspberry Pi via the general-purpose input-output pins. It plays the alarm sound supplied by the model and alerts the driver before a possible mishap.

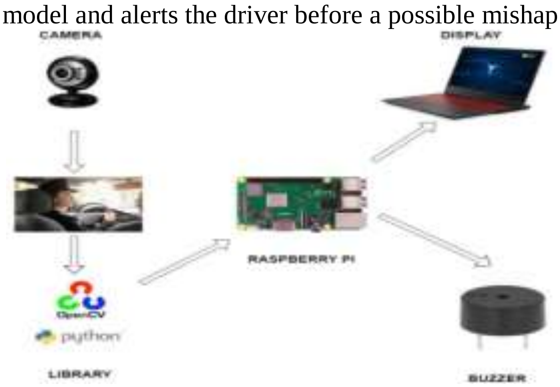

Figure 1. System Architecture

- Raspberry Pi: Raspberry Pi is a single-board computer, which provides all the input/output ports and is responsible for the processing and implementation of algorithms written in the Python programming language. It proves to be the most suitable processing unit for compact systems.
- OpenCV: It is a Python library used for face detection and is combined here with Haar cascades. Using the OpenCV library, we control the video stream input and perform the necessary functions on each frame.
- GPIO: The Python library used to facilitate commands to and from the Raspberry Pi. Here, the model uses the GPIO library to send the alarm signals to the buzzer, when the driver is detected to be drowsy.
- Display: The display is an output device, which is connected to Raspberry Pi and acts as an interaction medium between the Raspberry Pi and users/developers. We use the display module in our system to manage and validate our proposed model.

IV. IMPLEMENTATION

The proposed system uses Haar Cascades to detect objects in real-time, in this case, the face and eyes of the driver. The model uses libraries such as OpenCV, Dlib and GPIO to facilitate the working of the software and the input/output of the hardware used. The program initially defines two constants- EYE_AR_THRESH for the eye aspect ratio to indicate a blink and EYE_AR_CONSEC_FRAMES for the number of consecutive frames the system will monitor to detect drowsiness. Next, we initialize the frame counter 'COUNTER' to 0 and a boolean 'ALARM_ON' to OFF that tracks the number of frames for open/closed eyes and the status of the alarm. Now, we initialize the Dlib's HOG-based face detector and then create the facial landmark predictor using the facial features data provided in the facial_features.DAT file. Using this predictor, we extract the indexes of the facial landmarks for each eye with the help of the Imutils library.

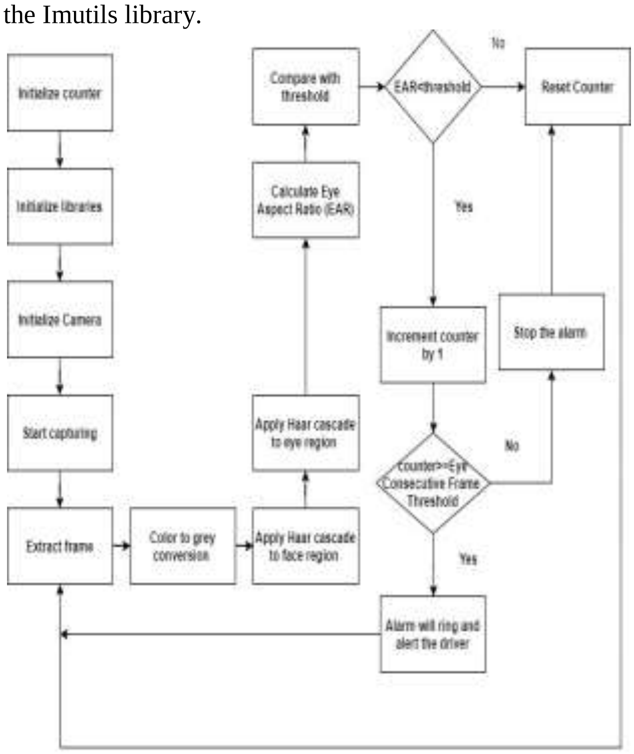

Figure 2. Flowchart of the algorithm

Now, we start the video stream thread and loop over each frame/ snapshot of the video.

The model resizes the frame and detects faces in the grayscale frame. Next, we extract the left and right eye coordinates from the facial landmarks of the face region and compute the eye aspect ratio for each eye. The Eye Aspect Ratio or EAR is calculated in three steps:

- The two pairs of vertical eye landmarks are taken and the Euclidean distance between them is computed by:

V1 = dist.euclidean(eye[2],eye[6])
V2 = dist.euclidean(eye [3],eye[5])

- The horizontal eye landmarks are taken and the Euclidean distance between them is computed by:

H = dist.euclidean(eye[1],eye[4])

Finally, the eye aspect ratio is calculated by (V1 + V2) / (2.0 * H) as shown by Fig.3.

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

Figure 3. Calculation of the Eye Aspect Ratio

Where p2 = eye[2], p6 = eye[6], p3 = eye[3], p5 = eye [5], p1 = eye[1] and p4 = eye[4].

This eye aspect ratio determines whether the eyes are open or closed, as demonstrated in Fig.4. The EAR is approximately constant while the eye is open, but will rapidly fall to zero when a blink is taking place and hence is used to detect a blink in a single frame. Using this simple equation, we can also avoid image processing techniques and rely on the ratio of eye landmark distances to determine if a person is blinking.

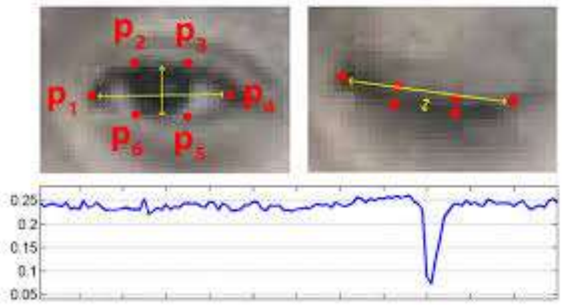

Figure 4. EAR used to determine openness of the eye

Now, we average the eye aspect ratio together for both eyes ((leftEAR + rightEAR) / 2.0) and compute the convex hull for the left and right eye. We mark them in green if the eyes are open or in red if the eyes are closed.

A condition statement checks if the eye aspect ratio is below the blink threshold, and if so, it increments the blink frame counter. If the eyes detected are closed in a consecutive of 5 frames, then we sound the alarm. Otherwise, we reset the counter back to 0. Here one iteration for a frame is completed, and we loop back to the next frame and repeat the same procedure for the new frame till the user closes the video console.

V. RESULTS AND DISCUSSION

Upon running the program, the dataset trains the model using the facial features of a human face, and it detects the eyes of the driver in real-time. The top right corner of the screen monitors the EAR or the Eye Aspect Ratio. This ratio determines the "openness" of the eyes and drops below 0.17 (the threshold value to determine the drowsiness of the system) when the system detects closed eyes. Each frame calculates its EAR, and a counter variable keeps track of the number of continuous frames where the eyes of the driver are closed. As soon as the counter reaches the threshold value of the model, in this case, five continuous frames, the driver is considered to be drowsy, and the raspberry pi sends an output signal to the alarm system using the GPIO library.

Figure 4. Model detects open and closed eyes based on the Eye Aspect Ratio

The camera module continuously monitors the video in real-time, and the driver is alerted by the sound played by the alarm system if found drowsy.

VI. CONCLUSION

The proposed system helps the driver to stay awake while driving and minimizes sleep-induced accidents. The model suggested, has proved to be efficient as the Raspberry Pie facilitated with the camera module is one the most compact and economic system that is adaptable for different motor vehicles.

The buzzer is responsible for alerting the driver by passing sound signals, which efficiently awakens the driver in real-time to avoid road accidents. The eye-detection using the Haar cascade classifier by calculating the Eye aspect ratio also reduces the false eye detections, a problem faced by the models using only the OpenCV library. Hence, the false positives are negligible, which increases the capability of the system. The Eye aspect ratio of consecutive frames will help us to remove those negligible errors and calculate drowsiness effectively. In the future, adding a high-resolution camera can ensure the proper functionality of the system in extreme light scenarios. Further, we can upgrade to a more powerful single-chip computer, such as LattePanda which have higher processing power and can thus minimize the lag suffered by the Raspberry Pi, to ensure faster alerts in real-time.

REFERENCES

1. Krajewski J, Sommer D, Trutschel U, Edwards D, Golz M. Steering wheel behavior based estimation of fatigue. The fifth international driving symposium on human factors in driver assessment, training and vehicle design 2009;118-124.
2. K. Fagerberg. Vehicle-based detection of inattentive driving for integration in an adaptive lane departure warning system Drowsiness detection, M.S. thesis, KTH Signals Sensors and Systems, Stockholm, Sweden, 2004
3. Savas, B. K., & Becerikli, Y. (2018). Real Time Driver Fatigue Detection Based on SVM Algorithm. 2018 6th International Conference on Control Engineering & Information Technology (CEIT). doi:10.1109/ceit.2018.8751886.
4. Danisman T, Bilasco IM, Djeraba C, Ihaddadene N. Drowsy driver detection system using eye blink patterns. Machine and Web Intelligence (ICMWI) IEEE 2010;230-233.
5. Donahue J, Anne Hendricks L, Guadarrama S, Rohrbach M, Venugopalan S, Saenko K, Darrell T. Long-term recurrent convolutional networks for visual recognition and description. IEEE conference on computer vision and pattern 2015;2625-2634.
6. Viola, P., Jones, M. Rapid object detection using a boosted cascade of simple features.Computer Vision and Pattern Recognition, IEEE Computer Society Conference.2001;I:I.

AUTHORS PROFILE

Shivani Sheth , Currently a final year computer science and engineering, B.Tech student in SRM Institute of science and technology. She has worked on building and optimizing the algorithm in the project, and has also compared different algorithms to find the suitable one compatible with Raspberry Pi and accessories. A machine learning enthusiast, she wishes to pursue her Masters in Artificial Intelligence and Data Science.

Aditya Singhal, Currently a final year computer science and engineering, B.Tech student in SRM Institute of science and technology. He is a diligent student and has immense interest in the field of research. Research in artificial intelligence field really excites him and wishes to do more such works and gain further knowledge and recognition relating to the subject in future.

Dr. V. V. Ramalingam, received Post graduate degree in Master of Computer Applications from Bharadhidasan University (2000), M.Phil Degree in Computer Science from Periyar University (2007) and M.Tech Degree in Computer Science from S.R.M University (2012) and completed Ph.D in Bharathiar University, Coimbatore. He also published twenty papers in International Journals and Conferences. His research interest is in the area of Data Mining using Machine Learning Approach.