International Research Journal of Education and Technology
Peer Reviewed Journal
ISSN 2581-7795

RASPBERRY PI BASED DROWSYNESS DETECTION FOR COMUTER VISION AND EYE ASPECT RATIO

AUTHOR NAME PRAVIN KUMAR S, KOWSICK N, SIVASANKAR S AND Mr. CHOZHARAJAN P

Dept of ECE, Bannari Amman Institute of Technology Associate Professor,Dept of ECE,Bannari Amman Institute of Technology

---***---

Abstract - Driving while fatigued is a major factor in traffic accidents all around the world. This research proposes a Raspberry Pi-based Drowsiness Detection System that uses computer vision and eye aspect ratio (EAR) to track the driver's attentiveness in an effort to increase road safety. The device is simple to install in a variety of cars since it combines a cheap Raspberry Pi single-board computer with a common USB camera. The suggested system tracks and analyzes the driver's facial features using computer vision techniques, paying particular attention to eye movements and closures. One important indicator for assessing tiredness is the ocular aspect ratio. The technology can accurately detect tiredness by continuously observing the driver's eyes, which includes slow blinks and prolonged eye closures.

Key Words*:* Real-time Accident Prevention Monitoring

1. INTRODUCTION

Driving while fatigued is a common issue that puts lives at risk by increasing the likelihood of collisions, injuries, and fatalities. Drowsiness or sleepiness, which affects a driver's alertness, is a significant factor in numerous traffic accidents. We have created a Raspberry Pi-based Drowsiness Detection System that makes use of computer vision and eye-aspect ratio (EAR) analysis to address this pressing issue and improve road safety. This system's primary focus is on tracking the motions and closures of the driver's eyes as it continuously scans their facial features. It is made into an affordable and quickly deployed solution for both personal and commercial cars by utilizing a low-cost Raspberry Pi single-board computer and a common USB camera.

1.1 System Overview

In this section, provide a brief but comprehensive overview of your Raspberry Pi-based Drowsiness Detection System. Explain the main objectives, the context in which the system will be used, and its primary functions. This serves as an introduction to the project, helping readers understand what the system is designed to achieve. Mention the use of computer vision and the Eye Aspect Ratio (EAR) as essential components for drowsiness detection.

1.2 Core Technology and Methodology

Under this subheading, delve into the core technology and methodology of your system. Explain in detail how computer vision techniques are employed to process the video feed, detect facial landmarks, and calculate the Eye Aspect Ratio

2. MATERIAL

Figure 1 displays the basic block diagram for driver drowsiness detection based on image processing. It consists of three key stages, i.e. input, process and output. The key idea of this system is to conduct real-time video monitoringon the driver's face from a camera and is able to accurately measure the level of driver drowsiness.The input stage includes a Raspberry Pi camera sensor to take a video of the eyes and the face of the drivers. The process stage includes Raspberry Pi 3 software using Open CV and Python for image processing using Haar Cascade and Eye Aspect Ratio algorithms. Output is the identification of drowsiness and buzzer rings as a way to warn the driver.Also, alert is send to the driver’s acquaintances through SMS and email. The location of the Pi camera in the vehicle is on

International Research Journal of Education and Technology
Peer Reviewed Journal
ISSN 2581-7795

the dashboard infront of the steering wheel. The driver will not be disturbed as the prototype in a small size.

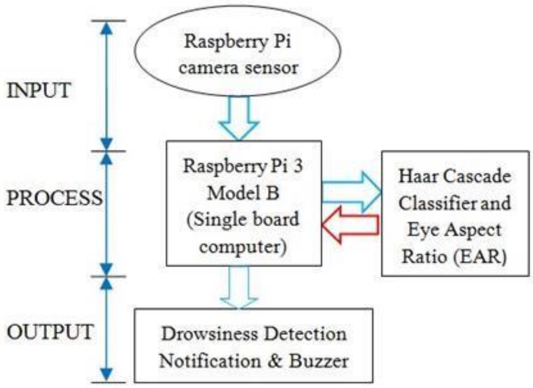


FIGURE.1

In order to start the execution of the program, the following libraries, such as NumPy, Open CV, and pi camera, need to be installed. The Open CV is released under a BSD license and is free for academic and commercial use. It has C, C++, Python, and Java interfaces, and it supports Windows, Linux, Mac OS, IOS, and Android. It is designed for computational efficiency and has a strong focus on real- applications. It is an open-source computer vision project that aims to provide a platform for the development of computer vision algorithms with a collection of libraries and applications. It provides I / O libraries for easy acquisition and manipulation of video data from multiple camera inputs.

3. METHOD

3.1 Image Processing :

Face and eye detection is a very vital and challenging issue in the field of image processing. It is also a crucial step in the recognition of the face. Open Source Computer Vision Library (Open CV) is used for the implementation of the Haar Cascade Classifier. In this project, the driver's sleepiness detection requires a video sensor to detect the eyes of the drivers. The driver's drowsiness level can be further determined by checking the eye blink rate. Figure 4 shows the methods used to detect face and eye, including eye blinks using the Haar Cascade Classifier and the Eye Aspect Ratio, respectively. Four key steps must be carried out in the Haar Cascade Classifier. The steps are Haar Feature, Integral Image, AdaBoost, and Cascade Classifier. Besides Eye Aspect Ratio is used to measure eye blinks (open and closed eyes) using a ratio formula based on both the width and height of the eyes.

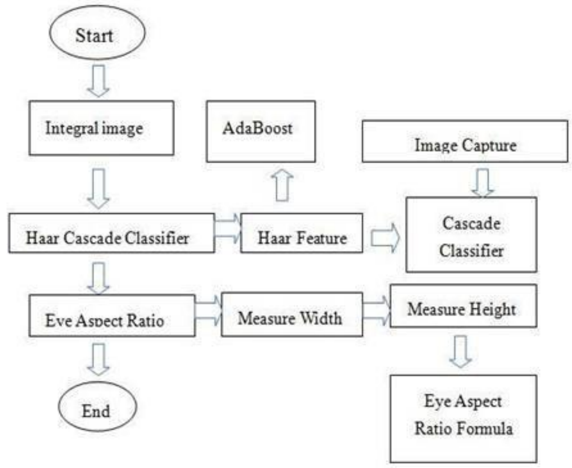


Figure 2. Methods involving the detection of face and eye blinks using the Haar Cascade classifier and the Eye Aspect Ratio, respectively

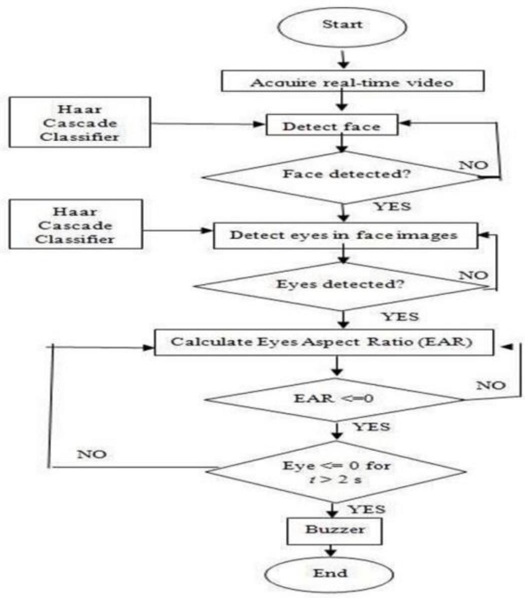


Figure 3. Overall face or eye including EAR detection flowchart

Figure 5 displays the general flow chart of the face and eye, including the EAR detection when the system is operated. The capturing of real-time video from the camera is performed by the raspistill command. As explained above, the Haar Cascade Classifier is used to detect face and eye images, while the EAR formula is used to detect eye blinking. Finally, during this prototype, the buzzer is used as a simple mechanism to warn the driver when the output EAR is approximately to zero for 2 seconds.

4. Eye Aspect Ratio (EAR)

After the driver's face is identified, the driver's drowsiness level determination is based on the eye blink rate. The Eye Ratio (EAR) formula proposed in [26] is in a position to detect an eye blink using a scalar value. For example, if the driver blinks his eyes more frequently, it means that the drivers are in a state of drowsiness. It is therefore important to accurately detect the shape of the eye to determine the frequency of blinking of the eye. From the landmarks identified in the face image, the EAR is used as an approximation of the state of eye openness. For each video frame, the outlines of the eyes are identified between the height and the width of the eyes that have been computed. The eye aspect ratio can be defined by Equation (4).

$$EAR = \frac{|p2 - p6| + |p3 - p5|}{2|p1 - p4|} \qquad (4)$$

.

Equation (4) demonstrates the eye aspect ratio formula where p1 to p6 is the 2D landmark spot. P2, p3, p5, and p6 are used to calculate the height, while p1 and p4 are used to measure the width of the eyes in meter (m) as shown in Figure 9(a). The eye aspect ratio is a fixed value when the eye is opened but sharply falls approximately to 0 when the eye is closed as shown in Figure 4(b).

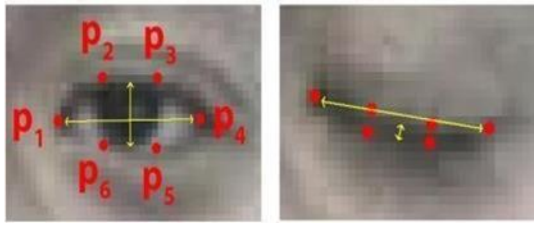


FIGURE 4

(a) Open eye and (b) Close eye with landmarks

When the eyes are open, the value of EAR is value greater than zero but when the eyes are closed, the EAR value falls rapidly to be approximate to zero.

5. DISCUSSION:

Figure 10 displays the Raspberry Pi operating setup with User Interface Development and the Raspberry Pi camera sensor interface. The main aim of this project is to investigate the implementation of Haar-Cascade and EAR algorithms in the Raspberry Pi environment. A number of five subjects were selected for the experiments designed and informed consents were obtained. A brief description of the procedures involved prior to the experiments was given to all participants. Experiments have been performed to analyze the efficiency of the Haar Cascade classifier and the EAR algorithms.

5.1. Real-Time Video Acquisition from Pi Camera:

The first experiment was to see whether or not real-time video could provide great quality video images. The camera was able to capture high quality video of the driver.

5.1. Determine the Haar Cascade Classifier for recognition of face and eyes:

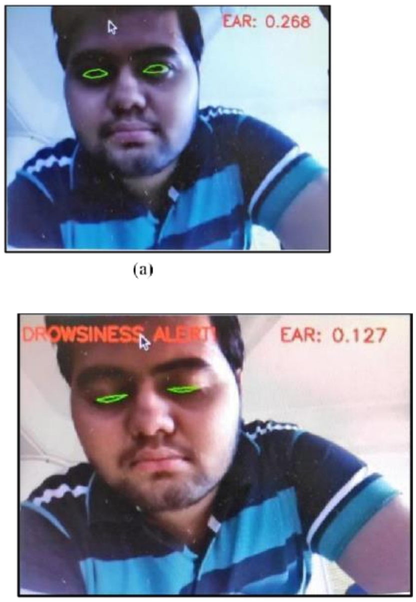
The second and third experiments were to determine the efficiency of the Haar Cascade Classifier to identify eyes and face by asking the driver to take a seat in various positions while wearing glasses and shades, respectively. Positions of the subject to be observed as if standing upright, the head turning and rotating in both directions, from left to right. In addition, participants were asked to wear spectacles and shades to determine whether or not the Haar Cascade Classifier could still detect the shape of the eye. These results showed that the Haar Cascade classifier was unable to identify the face and eyes when the subjects were not in the posture of sitting straight to thecameraor by wearing glasses. The resultant output will be affected by any movements made by the driver.

5.1. Determine eyes open and close using EAR formula:

The fourth experiment was to determine the EAR formula by evaluating the detection of eye landmarks. Based on the formula, it can be shown that if the EAR value is suddenly dropped, the driver may close his/her eyes. The EAR was determined on the basis of the equation (4) for each consecutive video frame and the EAR threshold was set in the code. The numerator of this equation calculates the distance between vertical eye landmarks while the denominator calculates the distance between horizontal eye landmarks.

5. RESULT AND ANALYSIS:

Results and analysis are provided in this segment. The key results of the Haar Cascade algorithm and the EAR formula are analysed on the basis of the various experimental studies. The EAR was determined on the basis of the equation (4) for each consecutive video frame and the EAR threshold was set in the code. The numerator of this equation calculates the distance between vertical eye landmarks while the denominator calculates the distance between horizontal eye landmarks. The EAR values for both eyes were then determined and the average EAR values for both eyes were shown on the screen. As described above, when eyes closed, the EAR result would be approximately 0 while, when eyes open, the EAR is always any number that is greater than 0. Figure 11 demonstrates the real-time measurement of the EAR during open and closed eyes. It is also seen that the EAR value appeared on the screen as seen in Figure 6(a) which was 0.268 when the eyes opened (not drowsy). Meanwhile, the EAR value was small (0.127) when the eyes closed (indicate sleepiness). The notification of drowsiness warnings is also shown in Figure 6(b).



(a) (b)

Figure 6. Example in real time EAR calculation during (a) eyes opened (EAR=0.268) and (b) eyes closed (EAR=0.127). A notification of drowsiness detection was popped out in the screen during eyes closed.

As the driver is detected as drowsy after a sudden drop in the EAR value of the eyes, the system sends text message and email to the other person as shown in Figure7.

Figure 7. Text message received on mobile number

Table 1. The average and estimated value of EAR during eyes opened and closed from five subjects

ects	EAR during eyes open	EAR during eyes closed
1	0.268	0.127
2	0.321	0.083
3	0.315	0.154
4	0.217	0.119
5	0.220	0.112
Average	**0.269**	**0.119**

Table 1. The standard eye-opening and closing EAR value of five subjects Table 1 shows the measured average EAR value of five subjects. The standard eye ratio is 0.269 and 0.119 when the eyes are opened and closed, respectively. From this experiment, it is often observed that zero average EAR value could not be achieved while eyes closed. However, based on the EAR test, it is also concluded that if the EAR value is unexpectedly decreased, it means that the driver has possibly closed his eyes. As a consequence, the decreasing value of the average EAR from 0.269 to 0.119 suggests a sign of eye blinking or eye closeness as seen in Figure 6.

3. CONCLUSIONS

In conclusion, the implementation of Raspberry Pi-based Drowsiness Detection System using Computer Vision and Eye Aspect Ratio (EAR) analysis represents a significant advancement in enhancing road safety. This project has successfully addressed the critical issue of driver drowsiness by continuously monitoring a driver's alertness during their journey

ACKNOWLEDGEMENT

The Eye Aspect Ratio (EAR) and computer vision are used in a Raspberry Pi sleepiness detection system to track driver alertness. Drivers can be warned if the technology detects symptoms of tiredness by examining facial characteristics and EAR. This technology is essential for improving traffic safety and averting accidents brought on by drowsy driving

REFERENCES

[1] [1] A. Malla, P. Davidson, P. Bones, R. Green and R. Jones, "Automated Video-based Measurement of Eye Closure for Detecting Behavioral Microsleep", in 32nd Annual International Conference of the IEEE, Buenos Aires, Argentina,2010.

[2] [2] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of SimpleFeatures", in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001.

[3] [3] R. Lienhart and J. Maydt, "An Extended Set of Haar-like Features for Rapid Object Detection", in Proceedings of the IEEE International Conference on Image Processing, 2002.

[4] [4] OpenCV. Open Source Computer Vision Library Reference Manual, 2001.