INTERNATIONAL JOURNAL OF RESEARCH CULTURE SOCIETY ISSN: 2456-6683 Volume - 4, Issue - 5, May – 2020
Monthly, Peer-Reviewed, Refereed, Indexed Journal Scientific Journal Impact Factor: 5.245
Received on : 05/05/2020 Accepted on : 19/05/2020 Publication Date: 31/05/2020

Driver Drowsiness Detection Using Raspberry Pi

[1]**Vishal Patil,** [2]**Sagar Panigrahi,** [3]**Gurudatta Patil** [4]**Dr. Amol Pande,**
[1, 2, 3] B.E Computer Engineering, [4]Head of Department and Assistant Professor
[1, 2, 3, 4] Department of Computer Engineering / Datta Meghe College of Engineering, Airoli, Navi Mumbai
[1, 2, 3, 4] Mumbai University, Mumbai, India
Email - [1]vishalpatil1403@gmail.com, [2]sagarpanigrahi68@gmail.com, [3]patilgurudatta98@gmail.com
[4]amolpande69@gmail.com

Abstract: *The drowsiness of driver is one of the leading contributing factors to the rising statistics of accidents in India. Hence, this paper proposes the design and development of driver drowsiness detection based on image processing using the Raspberry Pi camera module sensor interfacing with Raspberry Pi 3 board. To achieve the project objective, the Haar Cascade Classifier algorithm is used for eye and face detection while the Eye Aspect Ratio (EAR) algorithm is used for eye blink (open and close) detection. From several experiments on five recruited participants, the results showed that the accuracy of the Haar Cascade classifier for detecting the eyes and faces was linked to the correct sitting posture (head must face the camera) and that the eyes must not be hidden with glasses or shades. Whereas the system measured average EAR values ranged from 0.119 (closed eyes) to 0.269 (open eyes). In addition, the Haar Cascade and EAR algorithms used on the Raspberry Pi platform have been successfully performed for image processing. The current board can be replaced with Raspberry Pi Touch Screen to reduce the hardware setup for potential development, and the physiologically dependent analysis can be implemented using alcohol and heart rate sensors.*

Key Words: *Drowsiness, Haar Cascade Classifier, Raspberry Pi, Eye Aspect Ratio, Open CV, Python.*

1. INTRODUCTION:

Accident statistics as published on the Ministry of Road Transport and Highways website showed a rise from 147,913 cases of death (2017) to 151,417 cases of death (2018). Driver drowsiness has become one of India's major causes of road accidents and it can lead to physical injury or worse; death. Drowsiness is a state in which the person is experiencing a decreased level of consciousness due to lack of energy. The word drowsy means the sensation of falling asleep [1]. Driver exhaustion due to long working hours, lack of sleep and medical conditions is a significant cause of a road accident [2-4].Driver exhaustion may be characterized by behavioral reaction such as blinking the eyes at high frequency and nodding the head. Another consideration is that people drink too much alcohol. Alcohol-related accidents may have comparable fatality levels with driver drowsiness related accidents [5]. Furthermore, driving for long hours such as commuting by highway may be one of the examples of task-related factors that lead to driver drowsiness [6]. Because of these factors, drivers and other road users can lead to a collision, as these factors have impaired driver behaviour, visual ability to make good decisions, and speed of reaction[7]. Researchers have attempted to recognize driver drowsiness by using the following methods, such as vehicle-based trials, behavioural factors and physiological measures [8]. New technologies for driver fatigue identification, such as the system implemented in the Driver's Assistant in Ford vehicles, exist nowadays [9]. It investigates fast steering motions, driving on lines separating lanes, braking or acceleration erratic and hasty. In addition, some researchers have been using virtual conditions to manipulate drowsiness. Experimental power, performance, low cost, protection and ease of data collection are the key advantages of using simulator driver [10].

Driving simulators can be categorised as low-level simulators, mid-level simulators and high-level simulators that provide a 360-degree, extensive moving base [11]. Another technology used to protect the driver against the drowsy driver is to monitor the car on the basis of driving performance metrics such as detection of lane departures, large lateral lane deviations and termination of steering corrections [12]. Eye blinking is one of the signs of driver fatigue and is a major contributor to road accidents. Detecting the frequency of eye blinks (open and close) is important for the driver to notice drowsiness. There are a number of previous projects that Implemented eye blink detection, for example, embedded in human operator vigilance[13], where it will be notified when the user has not been blinking for a long time when viewing on the screen. Another illustration is face recognition with eye blink detection to prevent dry eyes and to protect against blinking [14]. Another advanced system is the Optalert Alert Monitoring System (OAMS) that used infrared (IR) reflectance oculography, which is one of the tools for monitoring eyelid movement [15]. It is used for detection of driver drowsiness in mining, transportation industries, and pilot sleepiness detection in aviation for detection and monitoring them while driving [16-18]. The system uses a sensor mounted on a spectacle frame and an IR emitter to make measurements continuously of the eye blink from where the degree of drowsiness is derived. The design and implementation of face and eyes recognition used to calculate behavioural measures for driver drowsiness detection is proposed in this project by using two main components: the Raspberry Pi 3 and the Raspberry Pi 8 Mega pixels. The aim of this project is to implement the Haar Cascade Classifier algorithm for face and eye recognition, to use the Eye Aspect Ratio algorithm to detect eye blinks that indicate drowsiness, and to evaluate both algorithms in different scenarios using recruited disciplines. The scope of the project was outlined as follows in order to achieve the objectives. The Raspberry Pi 3 is being used as the primary microcontroller, while the Raspberry Pi 8 Mega Pixel camera sensor is being used to capture images of disciplines. The algorithms are encoded using Open CV in the python environment. The buzzer is being used as a basic feedback mechanism to warn the driver when the drowsiness is detected. Also, a message and email will be sent to a friend or family member of the driver with a drowsy state alert and will eventually help the driver in the event of an emergency or accident. The prototype consisting of Raspberry Pi and Pi Camera is mounted together and can be placed inside the car.

2. LITERARURE REVIEW:

Previously, several types of research have been pursued to implement the Haar Cascade Classifier and the Eye Aspect Ratio in their project. For example, the Haar Cascade Classifier was examined for training on the performance of the author's face and eyes in order to design efficient eyes and face Haar Cascade Classifier detectors before they were combined into the hierarchical system [19]. This algorithm was able to detect and locate the eye area based on a set of test images. On the other hand, further research has been carried out on the Eyes Aspect Ratio for the driving monitoring system [20]. The experiment was carried out with a video camera that captured image frames of the driver's face to be the input of the system. The objective was to monitor driver awareness through the use of face detection and facial aspects. Driver drowsiness detection through multi eyelid movement features based on the information fusion methodology — partial least square regression (PLSR)—to resolve the issue of close collinear association between eyelid movement features and thus forecast the propensity of somnolence. The predictive accuracy and robustness of the model thus developed are validated, which show that it provides a novel way of combining multi-features to enhance our ability to detect and predict sleepiness [21]. A research based on visual examination of the eye condition and the head position (HP) was provided for continuous monitoring of vehicle driver alertness. Many of the current methods to visual recognition of non-alert driving habits focus either on eye closing or head shaking angles to determine the driver's sleepiness or distraction level. The help vector machine (SVM) classifies the series of video segments to warning or non-alert driving incidents [22]. Also, a new way of evaluating the driver's facial expression using Hidden Markov Model (HMM) dynamic simulation to identify sleepiness. The algorithm was applied using a virtual driving rig. The experimental findings confirmed the feasibility of the proposed procedure [23]. From the kinds of literature explored above, these initial attempts to recognise driver drowsiness is only found in a specific brand car and can be seen that the most of the drowsiness detection was relied on the driving performance measures and not on the driver's behavioural reactions when the somnolence the drowsiness In addition, there was no mention of the types of feedback to the driver to respond to the potentially dangerous driving performance indicators.

3. MATERIALS:

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

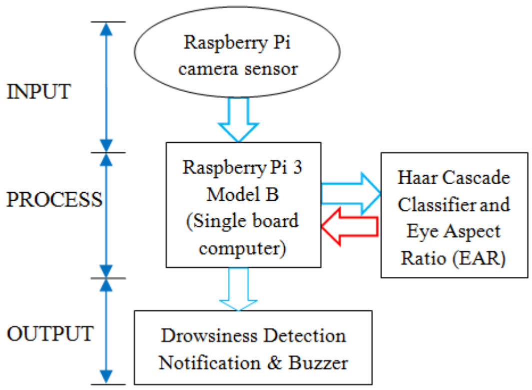


Figure 1. Block diagram of the device implemented

3.1. Hardware Apparatus:

The key tools for designing prototype-based image processing are the Raspberry Pi 3 Model B and the Raspberry Pi 8 Mega Pixel camera sensor. Figure 2 shows the Raspberry Pi 3 Model B, built by the Raspberry Pi Foundation. Although retaining the common board style, the Raspberry Pi 3 Model B brings a more powerful processor that is 10 times faster than that of the Raspberry Pi first generation. It is also fitted with wireless LAN and Bluetooth networking, making it the perfect solution for strong wired designs.

Figure 2. Raspberry Pi 3 Model B

In the meantime, Figure 3 shows the Raspberry Pi camera sensor, which is equipped with a high-quality 8-megapixel Sony IMX219 image sensor. It is capable of capturing 3280 x 2464 pixel images and support 1080p30, 720p60 and 640x480p60/90 videos. Its small size and light (with a weight of just over 3 g) make it suitable for portable applications. It can be connected to the Raspberry Pi via a short ribbon cable.

Figure 3. Raspberry Pi 8 Mega Pixel camera sensor

The Raspbian Operating System is a free, Debian-based operating system optimized for Raspberry Pi hardware. It includes a collection of simple programs and utilities running the Raspberry Pi. Python is a programming language that can be used to develop a computer program for Raspberry Pi when IDLE is an application that generates and runs Python programs. Python IDLE software with Open CV computer vision extensions is used to code the Haar Cascade and EAR algorithms. 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.

4. METHOD:

4.1. 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 close eyes) using a ratio formula based on both the width and height of the eyes.

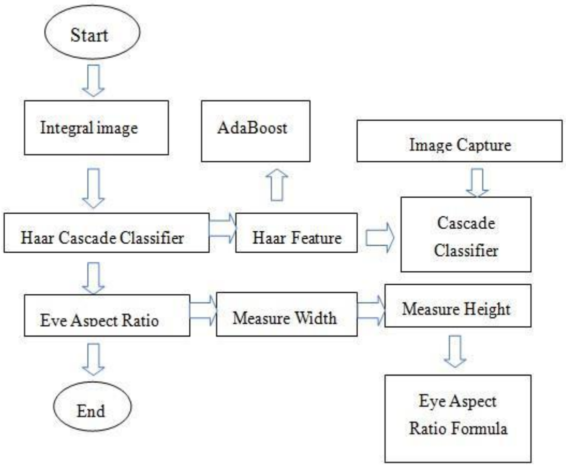


Figure 4. Methods involving the detection of face and eye blinks using the Haar Cascade classifier and the Eye Aspect Ratio, respectively

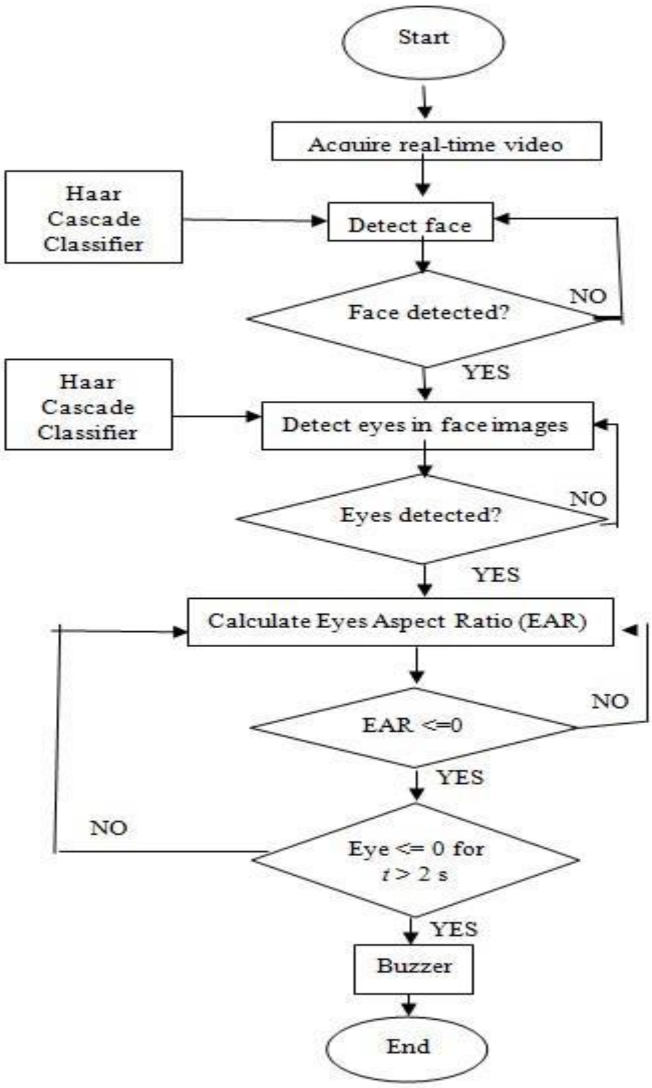


Figure 5. Overall face or eye including EAR detection flowchart

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.

4.2. Haar Cascade Classifier:

The Haar Cascade Classifier is one of the few object recognition methods capable of detecting faces. It offers high-speed computing depending on the number of pixels within the rectangle and not depending on the pixel value of the image. This method has four steps to detect an object, namely a Haar-like feature, an Integral Image, AdaBoost Learning and a Cascade Classifier [24]. Haar characteristics are the core component of the Haar Cascade Classifier for facial detection.The presence of a feature in the given image can be detected using Haar feature. Each feature results in a single value determined by the aggregate of the pixels under the black rectangle.The Haar-like feature is a rectangular feature that provides a specific indication of the image for rapid face detection [25]. Figure 6 displays the samples of common haar-like features.

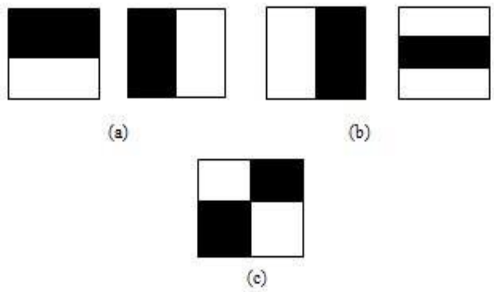


Figure 6. (a) Edge feature, (b) Line feature and (c) Four-Triangle feature

In order to obtain the object detection value, the Haar-like feature value was calculated using an integral image. It starts scanning the image to detect the face from the top left corner and ends the process of face detection at the bottom right of the image to detect the face from the image as shown in Figure 7(a). Integral image could measure values accurately and instantly by generating a new image presentation using the region value previously scanned with specific Haar-like features as shown in Figure 7(b). The value at any point (x, y)is the summed table of the aggregate of all the pixels above and the left of (x, y) inclusive as shown in Equation (1).

$$I(x, y) = \sum_{\substack{x' < x \\ y' < y}} i(x', y') \qquad (1)$$

Where i(x, y) is the value of the pixel at (x, y) and I(x, y) is the total amount of the integral values of the pixel. The value of the integral image, I(x, y) is obtained by the sum value of the previous index, starting from the top left to the bottom right. In addition, the summed-area table can be computed efficiently in a single pass over image as value in the summed-area table at (x, y) in Equation (2) [25].

$$I(x, y) = i(x, y) + I(x, y-1) + I(x-1, y1) + I(x-1, y-1) \qquad (2)$$

	1	2	3	4
1	3	7	7	3
2	1	3	3	1
3	5	9	9	5
4	3	6	6	3

(a)

	1	2	3	4
1	3	10	17	20
2	4	14	24	28
3	9	28	47	56
4	12	37	62	74

(b)

Figure 7. (a) Input image and (b) Integral image [25]

If the summed-area table has been computed, measuring the number of intensities for each rectangular area requires precisely four array references, no matter the world size. Equation (3) indicates the sum of i(x, y) above the rectangle of A, B, C, and D.

$$\sum_{\substack{x0<x<x1 \\ y0<y<y1}} i(x,y) = I(D) + I(A) - I(B) - I(C) \qquad (3)$$

The description in Figure 8 shows the calculation outline of the sum within the summed-area table structure having A=(x0, y0), B=(x1, y0), C=(x0, y1) and D=(x1, y1).

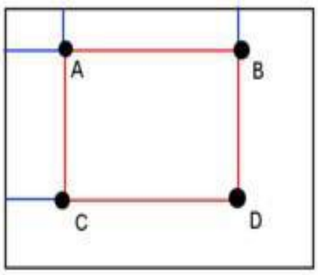


Figure 8. Description of the calculation of the sum in the summed-area data table

The value measured using an integral image will then be compared to the threshold value of the specific features given by AdaBoost. This should be done in order to define possible features, since not all features were appropriate for the particular object detection. AdaBoost blends the possible features called weak classifiers to become solid classifiers. The cascade classifier can be split into two strong classifiers and weak classifiers. Weak classifier means less accurate or irrelevant prediction, while a strong classifier means more precise prediction. A strong classifier built by AdaBoost can detect object level by level on a cascade. [26]

4.3. 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 9(b).

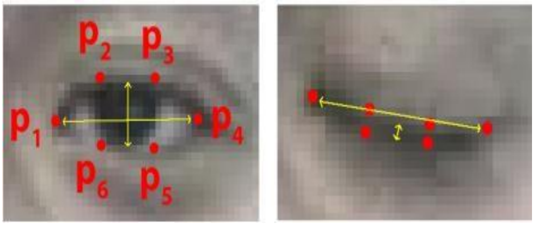


Figure 9 (a) Open eye and (b) Close eye with landmarks

$$EAR = \begin{cases} x > 0; \text{ eyes open} \\ 0; \quad \text{eyes close} \end{cases} \qquad (5)$$

Equation (5) demonstrates the EAR output range during eyes open and eyes close [27]. 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 display 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 analyse the efficiency of the Haar Cascade classifier and the EAR algorithms.

Figure 10. Raspberry Pi 3 Setup

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.

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

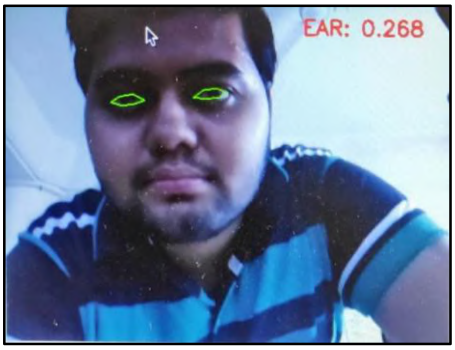
The second and third experiments were to determine the efficiency of the Haar Cascade Classifier to identify eyes and face by asking the driver to take a seat in various positions while wearing glasses and shades, respectively. Positions of the subject to be observed as if standing upright, the head 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.

5.3. Determine eyes open and close using EAR formula:

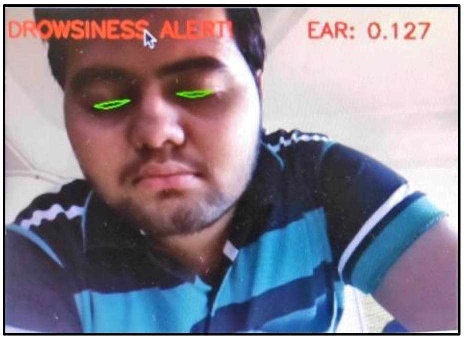
The fourth experiment was to determine the EAR formula by evaluating the detection of eye landmarks. Based on the formula, it can be shown that if the EAR value is suddenly dropped, the driver may close his/her eyes. The EAR was determined on the basis of the equation (4) for each consecutive video frame 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.

6. 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 11(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 11(b).



(a)



(b)

Figure 11. 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 Figure12.

Figure 12. Text message received on mobile number

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

Subjects	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 11.

Table 2. The average time taken by buzzer to respond when the eyes closed from five subjects

Subjects	Time taken by buzzer to respond after detect the eyes closed (s)
1	0.030
2	0.039
3	0.040
4	0.044
5	0.043
Average buzzer respond time	**0.039**

Meanwhile, Table 2 shows the time taken by the buzzer to answer when the eyes were closed of five subjects and the average time was 0.039 s. The quickest response time (less than 1 second) has shown that the buzzer is adequate to provide a clear feedback mechanism to warn the driver.

7. CONCLUSION:

In conclusion, all the objectives set at the beginning of the project have been successfully achieved. The development of image processing tasks based on Haar Cascade Classifier and Eye Aspect Ratio (EAR) algorithms on Raspberry Pi to predict sleepiness has been successfully developed and executed throughout this project. This project was conducted to propose a hardware platform prototype using the Raspberry Pi 3 Model B and the Raspberry camera sensor to operate image processing to detect driver drowsiness. These two main algorithms have been coded using Open CV and Python.In order to evaluate the performance of the Haar Cascade Classifier algorithm, five subjects were selected to perform a set of specific experiments. It pointed out that the Haar Cascade classifier was unable to find the face and eyes while the subjects shifted or wore glasses and shades. As for eye blink detection; the resultant EAR value was approximately 0.269 when the eyes opened, whereas when the eyes closed, the EAR was 0.119. Although it was not possible to obtain zero average EAR value, the sudden drop in average EAR indicates that the driver was in a state of sleepiness.As a mechanism to alert the driver, the warning message of drowsiness appeared on the screen, and the buzzer immediately rang in less than 1 second after the eyes had been closed.The device was able to send SMS and email to another subject after the driver's somnolence was identified.

REFERENCES:

1. A. Sahayadhas, K. Sundaraj, and M. Murugappan, (2012). “Detecting driver drowsiness based on sensors: A review,” Sensors (Switzerland), vol. 12, no. 12, pp. 16937–16953, doi:10.3390/s121216937
2. Fischer, Frida Marina, Claudia Roberta, De Castro Moreno, Flavio Notarnicola, and Fernando M Louzada, (2000). "Implementation of 12-hour shifts in a Brazilian Petrochemical Plant: Impact on sleep and alertness." 17, 521-537, doi:10.1081/CBI-100101062
3. Dawson, Drew, and Kirsty McCulloch, (2005). "Managing fatigue: It's about sleep." 365, 80, doi:10.1016/j.smrv.2005.03.002
4. Smolensky, Michael H, Lee Di, Maurice M Ohayon, and Pierre Philip, (2011) "Sleep disorders, medical conditions, and road accident risk." Accident Analysis & Prevention, 533-548, doi:10.1016/j.aap.2009.12.004
5. Pack, Andrew M, Eric Rodgman, Allan I Pacic, David F Dinges, and C William Schwab, (1995) "Characteristics the

driver of crashes attributed having fallen asleep to" 27, 769-75, doi:10.1016/0001-4575(95)00034-8
6. Williamson, Ann, David A Lombardi, Simon Folkard, Jane Stutts, Theodore K Courtney, and Jennie L Connor, (2011) "The link between fatigue and safety. " Accident Analysis & Prevention 43, 498-515, doi:10.1016/j.aap.2009.11.011
7. Saeed, Seyed, Hashemi Nazari, Ali Moradi, and Khaled Rahmani, (2012) "Original Article A Systematic Review of the Effect of Various Interventions on Reducing Fatigue and Sleepiness While Driving," Chinese Journal of Traumatology. - English Ed., vol. 20, no. 5, pp. 249–258, doi:10.1016/j.cjtee.2017.03.005
8. Sedighi, Zahra, Mohammad Ali, Alamdar Yazdi, Lora A Cavuoto, and Fadel M Megahed, (2017) "A data-driven approach to modeling physical fatigue in the workplace using wearable sensors," Applied Ergonomics., vol. 65, pp. 515–529, doi:10.1016/j.apergo.2017.02.001
9. Sałapatek, Damian, Jacek Dybała, Paweł Czapski, and Paweł Skalski, (2017) 'Proceedings of the Institute of Vehicles 3(112)', 41–48
10. Auberlet, Jean-michel, Florence Rosey, Sébastien Aubin, Patrice Briand, Marie-pierre Pacaux, and Patrick Plainchault, (2012)"The impact of perceptual treatments on driver's behavior: From driving simulator studies to field tests - First results. vol. 45, no pp. 91–98, doi:10.1016/j.aap.2011.11.020
11. N. Kaptein, J. Theeuwes, and R. Van Der Horst, (1996) "Driving Simulator Validity: Some Considerations," Transp. Res. Rec. J. Transp. Res. Board, vol. 1550, no. November, pp. 30–36.
12. Forsman, Pia M, Bryan J Vila, Robert A Short, Christopher G Mott, and Hans P A Van Dongen, (2013) "Efficient driver drowsiness detetion at moderate levels of drowsiness." Accident Analysis & Prevention 50, 341-350, doi:10.1016/j.aap.2012.05.005
13. Danisman, Taner, Ian Marius Bilasco, Chabane Djeraba, and Nacim Ihaddadene. (2010) "Drowsy driver detection system using eye blink patterns." Machine and Web Intelligence (ICMWI), International Conference on. IEEE, 230-33, doi:10.1109/ICMWI.2010.5648121
14. Pan, Gang, Lin Sun, and Zhaohui Wu, (2007) "Eyeblink-based anti-spoofing in face recognition from a generic web camera.1-8, doi: 10.1109/ICCV.2007.4409068
15. Johns, Murray W, Monitoring Von Augen Augenlid and Zusammenfassung Fragestel, (2007) "Monitoring eye and eyelid movements by infrared reflectance oculography to measure drowsiness in drivers." Somnology Sleep Research and Sleep Medicine, 242, 234-242, doi:10.1007/s11818-00 7-0311-y
16. Williamson, Ann, David A Lombardi, Simon Folkard, Jane Stutts, Theodore K Courtney, and Jennie L Connor, (2011) "The link between fatigue and safety." Accident Analysis & Prevention 43, 498-515, doi:10.1016/j.aap.2009.11.011
17. Corbett, Mark, and M A Corbett, (2015) "Science & Technology Watch: A Drowsiness Detection System for Pilots: Optalert Science and Technology watch a Drowsiness Detection System for Pilots." 80: 149-149, doi:10.3357/ASEM.21001.2009
18. Aidman, Eugene, Carolyn Chadunow, Kayla Johnson, and John Reece, (2015) 'Real-Time Driver Drowsiness Feedback Improves Driver Alertness and Self-Reported Driving Performance', Accident Analysis and Prevention, 81, 8–13, doi:10.1016/j.aap.2015.03.041
19. Kasinski, Andrzej, and Adam Schmidt (2010) , 'The Architecture and Performance of the Face and Eyes Detection System Based on the Haar Cascade Classifiers', 197–211, doi:10.1007/s10044-009-01505
20. Constantin, Leonte, Cristian Fosalau, Cristian Zet, and Daniel Petrisor, (2018) 'Driver Monitoring Using Face Detection and Facial Landmarks', International Conference and Exposition on Electrical And Power Engineering (EPE), 385–90.
21. Su, H., & Zheng, G. (2008). A Partial Least Squares Regression-Based Fusion Model for Predicting the Trend in Drowsiness. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, *38*(5), 1085–1092. doi: 10.1109/tsmca.2008.2001067.
22. Mbouna, R. O., Kong, S. G., & Chun, M.-G. (2013). Visual Analysis of Eye State and Head Pose for Driver Alertness Monitoring. *IEEE Transactions on Intelligent Transportation Systems*, *14*(3), 1462–1469. doi: 10.1109/tits.2013.2262098.
23. Tadesse, E., Sheng, W., & Liu, M. (2014). Driver drowsiness detection through HMM based dynamic modeling. *2014 IEEE International Conference on Robotics and Automation (ICRA)*. doi: 10.1109/icra.2014.6907440.
24. Viola, Paul, and Michael Jones, (2001) "Rapid object detection using a boosted cascade of simple features." Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. Vol. 1. IEEE, doi:10.1023/B:VISI.0000013 087.49260.fb.
25. P. Viola and M. M. J. Jones, (2004) "Robust Real-time Face Detection," Int. J. Computer. Vis., Vol. 57, No. 2, Pp. 137– 154.
26. Soukupova, Tereza, and Jan Cech. (2016) "Eye Blink Detection Using Facial Landmarks." 21st Computer Vision Winter Workshop, Rimske Toplice, Slovenia.
27. Kamarudin, N., Jumadi, N. A., Mun, N. L., Keat, N. C., Ching, A. H. K., Mahmud, W. M. H. W., … Mahmud, F. (2019). Implementation of Haar Cascade Classifier and Eye Aspect Ratio for Driver Drowsiness Detection Using Raspberry Pi. Universal Journal of Electrical and Electronic Engineering, 6(5B), 67–75. doi: 10.13189/ujeee.2019.061609.