Chapter 31
Fatigue and Drowsiness Detection System Using Artificial Intelligence Technique for Car Drivers

Mohd Azlan Abu, Izzat Danial Ishak, Hafiz Basarudin, Aizat Faiz Ramli, and Mohd Ibrahim Shapiai

Abstract Road traffic accident in Malaysia is a heavy concern in these days. Among the top factors of traffic accidents, the fatigue and drowsiness of drivers often times contributed to the increasing number of cases and fatality rate of accidents. This research aims to develop a computer vision system to detect such fatigue and drowsiness of the drivers and wake them up from the split-second nap. The implementation of this research is to develop a drowsiness detection system implemented in a compact development board to assist drivers to awaken from microsleep during driving on fatigue due to long driving hours and various other reasons. This research used a Raspberry Pi 4 along with the official Raspberry Pi camera module V2 and an active buzzer module as waking mechanism for the system. The development used and experimented on the Haar cascade classifier and Histogram of Oriented Gradient + linear Support Vector Machine in the effort of determining the best suitable model to be used for drowsiness detection in terms of speed and accuracy. Both models were run and tested to work properly. The implementation of the Haar cascade classifier produced the best performance in terms of speed and response time to detect drowsiness. On the other hand, the HOG + SVM had better accuracy when compared to the Haar cascade classifier even in low illumination. Having said that, the response time is significantly slower than Haar model which caused a problem regarding the

M. A. Abu (✉) · M. I. Shapiai
Malaysia-Japan International Institute of Technology, Universiti Teknologi Malaysia, Jalan Sultan Yahya Petra, 54100 Kuala Lumpur, Malaysia
e-mail: mohdazlan.abu@utm.my

M. I. Shapiai
e-mail: md_ibrahim83@utm.my

I. D. Ishak · H. Basarudin · A. F. Ramli
Universiti Kuala Lumpur British Malaysian Institute, 53100 Gombak, Batu 8, Jalan Sg Pusu, Selangor, Malaysia
e-mail: izzat.ishak07@s.unikl.edu.my

H. Basarudin
e-mail: hafizb@unikl.edu.my

A. F. Ramli
e-mail: aizatfaiz@unikl.edu.my

A. Ismail et al. (eds.), *Design in Maritime Engineering*, Advanced Structured Materials 167, https://doi.org/10.1007/978-3-030-89988-2_31

reaction time of drivers to react on time. To conclude, the Haar cascaded classifier is decided as the most appropriate model to be applied for the development of a drowsiness detection system.

Keywords Computer vision · Applied AI · Artificial intelligence · Eye aspect ratio · Embedded system

31.1 Introduction

Road traffic accident is a heavy concern all year long, ranging from different factors of traffic crashes that are often the cause of fatality, particularly in Malaysia. Regrettably, one of the top factors of traffic accidents includes the fatigue and drowsiness of drivers especially during long hours of drive as well as occupational driving scenario which is easily solvable by closing the eyes and take a rest from driving for a moment of time. The director the general of Health Ministry Datuk Dr. Noor Hisham Abdullah also stressed that drivers are not being on the road with the feeling of fatigue and tiredness as to avoid road accidents [1]. According to the Malaysian Institute of Road Safety Research (MIROS) in their annual statistical report [2], driver fatigue is one of the most critical factors to be concerned to reduce fatality due to traffic accidents as they cause more fatalities per case which is on par with the risky driving and speeding cases. The report also states that fatigue cases are recorded as the most prominent factors of crash occurrence which is 8.6% of total cases. The trend tends to be similar throughout the year-long duration.

The highlight of the proposed research is to monitor the eye movements of the driver's eyes whereas the subject is recognized as drowsy when the eyes are shut after even a short time. The way to achieve this is by implementing the facial landmarks to extract eye features from the face and afterwards calculating the eye aspect ratio between the height and width of the eyes. By referring to the paper "Real-Time Eye Blink Detection using Facial Landmarks" by Tereza Soukupova and Jan Cech [3], the concept is represented in Fig. 31.1 where the landmark points are determined at the corner, top and bottom for the height and width of the eye. According to the paper, the eye landmarks are detected for every frame while calculating the eye aspect ratio (EAR) to determine the state of the eyes using the formula given in Eq. (31.1), where

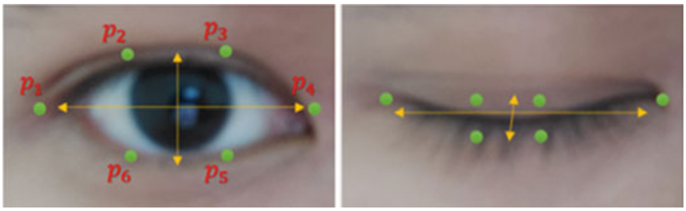

Fig. 31.1 Facial landmarks pinpoint the eye features

"P" are the 2D landmark locations [3].

$$\mathrm{EAR} = \frac{\|P_2 - P_6\| + \|P_3 - P_5\|}{2P_1 - P_4} \tag{31.1}$$

Equation (31.1) represents the average of eye aspect ratio on each eye because of the simultaneous blinking of both. When in open state, the EAR is typically constant due to the eye's minimal movements. While in closed state, the ratio becomes close to zero in which the values can be used to program for indicating shut eyes. Onto the subject of varying eye size, this however does not affect the reliance for this concept as the aspect ratio of open eyes of different individuals has the unnoticeable small variance to compute. With the use of this EAR concept, the system can be programmed to signal for alert from the driver's drowsiness through the person's shut eyes for more than at least a second.

31.2 Literature Review

This segment highlights the studies of components and elements considered as essential in constructing from a basic face detection system to an advanced and efficient one, utilizing recent technologies. The section also comprises of latest and recent studies involving the contributions, challenges and varying development of face detection systems. Several journals, articles and reports are referenced for the purpose of this research.

31.2.1 Large-Scale Human Detection Using HOG

Detecting humans and other living things is a challenging task to execute because of the wide variations of outward forms and postures they take on in images and videos. To add to that, difficult lightings and cluttered surrounding settings further complicate the task of detecting humans even on a smaller scale. A robust and efficient detection algorithm is needed to overcome the difficulties of filtering human forms from difficult backgrounds and illuminations.

The implementation of HOG for a large-scale human detection is experimented in the journal by Dalal and Triggs [8] using the pedestrian database as dataset, which consists of 509 training and 200 test images of pedestrians in city scenes. Among the sample images used, the human subjects tend to be upright, with addition to limited obstructions and a wide range of variations in appearances, lighting, backgrounds, postures and apparels in a crowded city spot. Basically, the HOG-based detectors greatly outperform that of other wavelet-based detectors especially the Haar-like wavelets that comes second after all other HOG-based detectors. The performances of final rectangular (R-HOG) and circular (C-HOG) detectors that utilize the SVM algorithm that possess similar low miss rates of detection contribute to the near-perfect accuracy of human detection using the HOG with linear SVM algorithm.

This paper proved that using the HOG gives very good results for person detection, reducing false positive rates by more than an order of magnitude relative to the best Haar-based detector. As stated by the authors [8], “although our current linear SVM detector is reasonably efficient by processing a 320 × 240 scale-space image (4000 detection windows) in less than a second there is still room for optimization and to further speed up detections it would be useful to develop a coarse-to-fine or rejection-chain style detector based on HOG descriptors”.

31.2.2 Face Detection and Face Recognition on Raspberry Pi

The combination used of both Haar-like features and HOG + SVM is precisely demonstrated in the paper by Deshmukh [10] where the Raspberry Pi 3 is used to create a security system that is trained to identify unknown individuals using the HOG + SVM algorithm in Python language.

The first and foremost step is the face detection using the Haar cascade classifier built in the OpenCV libraries to create a database of positive and negative images. From there, the HOG handles the normalization process of the images to result in better invariance towards changes in lightings, shadowing and black points. This helps for a better and effective training by the SVM algorithm. The SVM classifier then trains data to recognize the designated face from many faces and background objects.

The author stated in the journal [10] that the use of Raspberry Pi for a face recognition system can make for a lighter and lower power consumption of a security device, so it makes for a more convenient system than a PC-based face recognition. Due to the open-source nature, it is easier to program a software development on Linux because of wide availability of source codes and references in Python language. The analysis made by the authors revealed that the proposed system shows promising performance in face recognition while being able to be used for face detection even from poor quality images.

In contrast to the above discussion, the study by Gupta and team [11] makes use of the eigenface approach by using the principal component analysis (PCA) algorithm for face recognition process. This algorithm utilizes a mathematical procedure that transforms several possibly correlated variables into variables. The authors state that the eigenface approach used can aid in reducing the size of the database required for recognition. This conventional eigenface approach is implemented into the Raspberry Pi’s ARM Cortex for the purpose of face recognition using face recognition modules in Python programming.

31.3 Methodology

This section gives the insight and overview of details of the proposed research methodology to achieve the purpose of developing an effective drowsiness detection system,

The discussions and procedures included in this chapter are important to be achieved. The aim and objectives of the research in developing the drowsiness detection system will be discussed.

31.3.1 Block Diagram

The drowsiness detection system essentially comprises of the Raspberry Pi 4 board itself as the device, connected with the official Raspberry Pi camera module for eye scanning and a display to monitor the perspective of the camera for debugging, through its I/O ports. The eye features will be extracted upon detection of the facial landmarks, particularly the eye regions. The process is seen as a block diagram in Fig. 31.2 which contains the input, process and output of the system.

Input: Live feed obtained from the camera module will be taken as the input to be sent to the central unit.

Process: Left and right-hand eyes from the video stream will be detected. The dlib library from OpenCV will be utilized to deliver the information for extracting only the eye region from the facial landmarks. The camera continues to monitor the eye aspect ratio to be calculated for indications of shut eyes in a set period in seconds. The video stream of the camera module will be shown on the display in real-time mainly for monitoring and debugging purposes. Green outlines will appear around both eyes for eye aspect ratio calculations and drowsiness alerts will be indicated by a red sentence "drowsiness alert" on the screen.

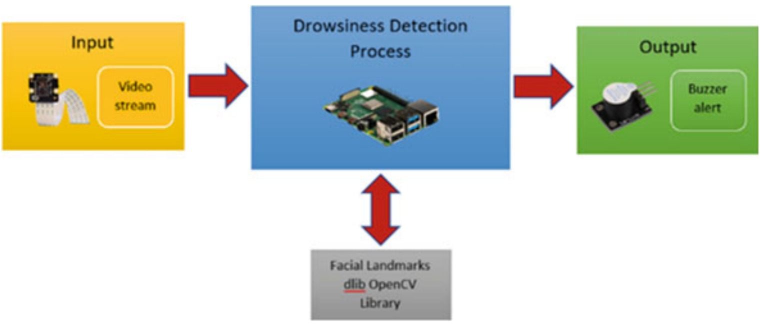

Fig. 31.2 Drowsiness detection system block diagram

Output: An active buzzer module will produce beep sounds to alert the driver upon drowsiness. The high frequency beep sounds serve the purpose of waking a person from the drowsiness.

The drowsiness detection will start with detecting any visible eyes from the camera video stream. Then, the drowsiness detection process will be executed by extracting the eye region from the facial landmarks referencing from the dlib OpenCV library. The system will then monitor the eye aspect ratio with range of 0.3 and below, in which once the ratio approximately below 0.3 is detected for a set time, the system will recognize as the driver being drowsy. Thus, a signal will be sent to the buzzer to ring the driver from drowsiness. Otherwise, the detection system will continue to monitor the eye ratios from the camera's perspective.

31.4 Result and Discussion

This section presents and discusses the result obtained from the program actual test run using a subject's eyes to detect drowsiness in real time. The figures presented in this chapter are the screenshots of the actual live video stream on different conditions.

31.4.1 Haar Cascade Classifier

Tests are run using the Haar cascade classifier to gauge the algorithm's performance involving the framerate per second (fps), response time (ms) and different conditions during the test run. The test was first running on normal condition to determine the performance on the fundamental level. Figure 31.3 shows the screenshot of real-time video feed of the subject with outlines around the eyes' shape.

Fig. 31.3 Video feed on normal condition

Fig. 31.4 Drowsiness alert

Figure 31.4 shows both screenshot of real-time feed without outlines of the shut eyes to test the detection process of the Haar algorithm. When the eyes are shut, the EAR rate has dropped from the threshold of 0.3 to an average of 0.21 within 2 s which is supposed to be the indication of the subject's drowsiness. The drowsiness alert indication then popped up on top left as well as the beeping sound of the buzzer to wake the subject. The fps rate is still maintained on the average of 13.5 which demonstrated a lesser lag in using the Haar cascade classifier in monitoring the eye features.

The findings showed favourable results using the method of eye blink detection by implementing the facial landmarks detector from the works of Soukupova and Cech [3]. However, instead of using HOG + SVM, this method used the Haar cascade classifier to compensate for the lesser processing power of the compact Raspberry Pi 4.

31.4.2 Hog + svm

In Fig. 31.5, the implementation of HOG + SVM on the Raspberry Pi 4 is successful. However, the recorded fps is substantially lower which is two times lower than that of applying the Haar cascade classifier. Although the drowsiness detecting succeeded as shown in Fig. 31.6 even in poor illumination, the response time to detect is significantly slower in contrast with the Haar cascade classifier, which is undesired.

In Table 31.1, it is shown that the average fps of implementing the Haar cascade classifier is two times higher than that of HOG + SVM. The response time for the Haar detector to detect drowsiness is also significantly faster at 1.12 s than HOG + SVM at 6.95 s. This makes the Haar cascade classifier substantially faster in processing detection on the Raspberry Pi 4 as well as putting less burden to the processor.

Fig. 31.5 HOG + SVM normal condition

Fig. 31.6 HOG + SVM detects in low illumination

Table 31.1 Comparison of Haar and HOG + SVM

	Haar cascade classifier	HOG + SVM
Average FPS	13.5	6.2
Response time (s)	1.12	6.95

Overall, the HOG + SVM has undeniably better accuracy than the Haar detector especially in the findings of detecting in dark and low illumination condition. But the response time posed a problem in terms of the response of the subject to wake from microsleep during driving. The subject might not have the time to react by the time the HOG + SVM detects drowsiness. Therefore, the Haar cascade classifier seemed to be the best fit specifically in this case of implementing on the small and compact Raspberry Pi 4.

31.5 Conclusion

Choosing the appropriate AI model is of utmost importance when developing a safety feature that intends to keep people from harm, not vice versa. A low performance drowsiness detection system could not possibly satisfy the minimum safety standards and therefore beats the purpose of the system. From the results, the HOG with SVM and Haar cascade classifier are tested to measure and gauge the performance of the models upon running the program in Raspberry Pi 4. It is observed that the Haar cascade classifier favours more in performance and detection speed in contrast with the HOG with SVM that puts more burden on the Raspberry Pi processing power and consequently reducing the detection speed of the system. Given the surrounding setting of the driver seat in the vehicle interior where the driver is positioned, it is decided that detection accuracy is less emphasized due to lesser faces and eyes are to be detected in a driver's seat, in which the use of the Haar cascade classifier is sufficient enough for the task. The detection accuracy upon implementing this model is fairly compensated with its detection speed offered. For future recommendation, the current camera module used in the research which is the official Raspberry Pi camera module lacks the ability to see clearly in dark and poor-light time of the day. The possible solution to overcome this is by swapping out the current camera with the Raspberry Pi NoIR camera module. This module not only gives everything the regular camera module offers, but also has the night vision capability. The NoIR that stands for "No Infrared" simply means it does not employ an infrared filter. This means that images and videos taken by daylight will look distorted on the colours, but in turn gives the ability to see in the dark with infrared lighting. The only concern is how well the detection performs with distorted colours and dim vision. This should be the task for future works progress.

The research sacrifices a slight bit of accuracy in return for the speed of detection by using the Haar cascade classifier model instead of the more accurate HOG with SVM. This is due to the demanding processing power from the HOG with SVM to process the detection with better accuracy. The current performance could potentially bring about undesired false positives and false negatives upon the detection process. Further research and tests should be made to find the best possible model to be implemented into the system that favours both speed and accuracy attributes.

Acknowledgements Author would like to acknowledge Malaysia-Japan International Institute of Technology, Universiti Teknologi Malaysia, Jalan Sultan Yahya Petra, 54100 Kuala Lumpur for the funding provided.

References

1. Bernama (2017) "Don't drive if you are fatigued, sleepy—health DG," http://english.astroawani.com/malaysia-news/dont-drive-if-you-are-fatigued-sleepy-health-dg-144224. Accessed 14 Jun 2020

2. Zainal AANS, Mohd FSA, Lamin F, Abdul MAR (2012) MIROS crash investigation and reconstruction annual statistical report 2007–2010
3. Cech J, Soukupova T (2016) "Real-time eye blink detection using facial landmarks," Cent Mach Perception, Dep Cybern Fac Electr Eng Czech Tech Univ Prague, pp 1–8. https://doi.org/10.1017/CBO9781107415324.004
4. Farley P (2019) "Face detection and attributes concepts—azure cognitive services | microsoft docs,". https://docs.microsoft.com/en-us/azure/cognitive-services/face/concepts/face-detection#face-landmarks . Accessed 16 Jun 2020
5. Dwivedi D (2018) "Face detection for beginners—towards data science," https://towardsdatascience.com/face-detection-for-beginners-e58e8f21aad9 . Accessed 15 Jun 2020
6. Vanderplas J (2016) "Python data science handbook," https://jakevdp.github.io/PythonDataScienceHandbook/ . Accessed 15 Jun 2020
7. Berger W (2020) "Deep learning haar cascade explained." http://www.willberger.org/cascade-haar-explained/. Accessed 15 Jun 2020
8. Dalal N, Triggs B (2018) "Histogram of oriented gradients for human detection," Proc 2018 5th Int Conf Bus Ind Res Smart Technol Next Gener Information, Eng Bus Soc Sci ICBIR 2018. https://doi.org/10.1109/ICBIR.2018.8391187
9. Frigerio A, Hadlock TA, Murray EH, Heaton JT (2014) Infrared-based blink-detecting glasses for facial pacing: toward a bionic blink. JAMA Facial Plast Surg 16(3):211–218. https://doi.org/10.1001/jamafacial.2014.1
10. Deshmukh SV et al (2017) Face detection and face recognition using raspberry pi. Ijarcce 6(4):70–73. https://doi.org/10.17148/ijarcce.2017.6414
11. Gupta I, Patil V, Kadam C, Dumbre S (2017) "Face detection and recognition using raspberry pi," WIECON-ECE 2016–2016 IEEE Int WIE Conf Electr Comput Eng, pp 83–86. https://doi.org/10.1109/WIECON-ECE.2016.8009092
12. Kwolek B (2014) "Face detection using convolutional neural networks and gabor filters," Lect. Notes Comput Sci 3696. https://doi.org/10.1007/11550822
13. Kamencay P, Benco M, Mizdos T, Radil R (2017) "A new method for face recognition using convolutional neural network," Adv Electr Electron Eng 15(4): 663–672. https://doi.org/10.15598/aeee.v15i4.2389
14. Reddy B, Kim Y H, Yun S, Seo C, Jang J (2017) "Real-time driver drowsiness detection for embedded system using model compression of deep neural networks." In IEEE computer society conference on computer vision and pattern recognition workshops. https://doi.org/10.1109/CVPRW.2017.59
15. Zhang K, Zhang Z, Li Z, Qiao Y (2016) "Joint face detection and alignment using multi-task cascaded convolutional networks," IEEE Signal Process Lett. https://doi.org/10.1109/LSP.2016.2603342