Real-Time Driver Alert System Using Raspberry Pi

Jie Yi Wong[1] and **Phooi Yee Lau**[2†], Non-members

ABSTRACT

Malaysia has been ranked as one of the countries in the world with the deadliest roads. Based on the reported statistics for 2016, a total of 7152 people died due to road accidents in Malaysia, and these accidents are often caused by human errors. Efforts to curb these accidents include increasing patrols by authorities and joint campaigns involving transport authorities and police but these efforts are not long-term as they only focus around accident hotspots nationwide. The Advanced Driver Assistance System (ADAS) aims to assist drivers and to increase car safety, not only through improving driving experience but also through taking into consideration the overall passenger safety. In recent years, driver's drowsiness has been one of the major causes of road accidents, resulting in severe physical injuries and deaths, resulting in significant economic loss. In this paper, a vision-based real-time driver alert system, aimed mainly to monitor the driver's drowsiness condition and distraction condition, is proposed. The real-time driver alert system consists of (1) a drowsiness warning system in the Driver Drowsiness Monitoring System (DDMS) module, and, (2) a Raspberry Pi development board, a camera, two LEDs, and a buzzer in the Raspberry Pi Deployment (RPD) module. The DDMS module focus on monitoring the driver's condition in a quad-step method: (1) detecting driver's face, (2) detecting eyes region, a (3) estimating the rate of eyes closure, i.e. drowsiness level, and (4) estimating the head pose, i.e. inattentiveness in driver, in order to monitor the driver's condition. The RPD module focus on (1) embedding the DDMS system into the Raspberry Pi, mounted with a Raspberry Pi camera with a buzzer, to capture the driver's condition, and (2) mounting the system in the car, to alert the driver by providing a medium loud beeping sound. Experimental results show that proposed system is practical and low-cost which could (1) embed the real-time drowsiness detection module, and (2) provide alert notification to the driver when the driver is inattentive, using a medium loud beeping sound, in real-time.

Manuscript received on January 26, 2019 ; revised on June 22, 2019 ; accepted on July 21, 2019.

The authors are with Universiti Tunku Abdul Rahman, Kampar, Malaysia. E-mail: wongjieyi@1utar.my[1], laupy@utar.edu.my[2]

†Corresponding author.

Keywords: ADAS, Drowsiness, Raspberry Pi, Computer-vision-based, Real-time

1. INTRODUCTION

About 1.24 million people die on the roads annually, with 20-50 million sustaining non-fatal injuries [1]. From 2009 to 2013, drowsy driving has caused a minimum of 72,000 accidents. Approximately 800 deaths and 41,000 injuries were caused by these crashes. However, these numbers are often underestimated and up to 6,000 fatal crashes each year may be caused by drowsy drivers [2]. Usually around 21% of road accidents are caused by driving in drowsy condition, which is one out of five road accidents. Based on the data from 180 different countries, this percentage is increasing every year. This certainly highlights the fact that across the world the total numbers of road traffic deaths are highly due to driver's drowsiness [3].

Drowsiness condition could reduce the driver's attentiveness while driving and impairs the eventual decision made by the driver. Drowsiness-related crashes are more likely to occur during the early morning, mid-afternoon and also late at night. There are a lot of fatigue signs which could represent drowsy drivers, such as a driver yawning repeatedly, a driver having difficulty focusing or keeping his/her eyes open, and a driver having trouble keeping his/her head up. Many systems have been developed recently [4–8] to track driver drowsiness condition. In 2007, Eskandarian and Mortazavi proposed to study the drowsiness conditions using a neural network-based truck-driving simulator. However, no real-time performance implementation was reported. In 2010, Vitabile et. al implemented a drowsiness detection system on an FPGA-system using (1) an infra-red camera installed, to stream real-time video, and (2) a buzzer, to alert the driver. However, the system focused only on the detection of eye pupil alone. In 2013, You et al. proposed an android-based system to warn drivers who drive recklessly, such as driving when the driver is sleepy and unfocused, using a visual alarm. Thia system implements on a smartphone using the front-facing camera to detect reckless driving and a rear-facing camera to track road conditions. In 2015, Wang et al proposed a system to monitor the state of the driver on a real-time network and share this information among all drivers in the cloud. Though the system has incorporated

Fig.1: *Simple illustration of how the DDMS works.*

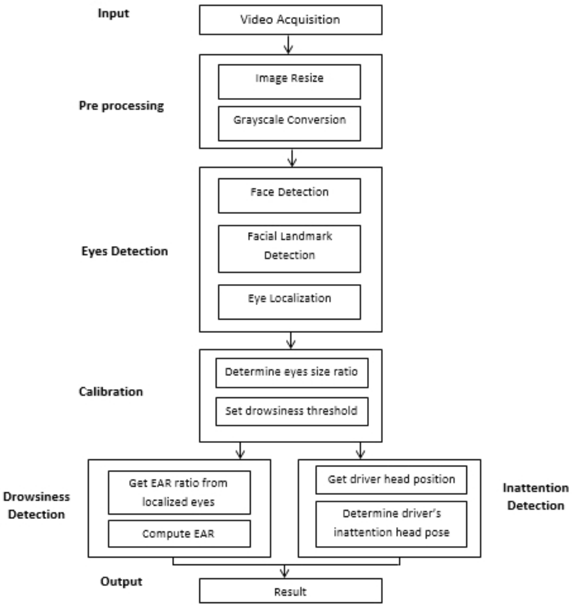


Fig.2: *Proposed DDMS Framework.*

the detection, communication and service module, there were no real-time implementation results presented. In 2016, Musale and Pansambal proposed to detect the driver's eye state using Raspberry Pi. However, the proposed system lacks on-the-road real-time implementation conditions. In 2018, Chang et al proposed a drowsiness-fatigue-detection system using wearable smart glasses. However, the wearable-based system could exhaust the eyes easily, when drivers wear it for a long period, in real-implementation scenarios. These previous works indicate that there is a lot of potential for a real-time driver alert systems in monitoring fatigue or distracted conditions in drivers.

The remainder of this paper includes; Section 3 that outlines the DDMS module; Section 4 that describes the RPD module; Section 5 that discusses the performance of the modules with analysis; and Section 6 that concludes the work.

The Advanced Driver Assistance systems (ADAS) technology has the potential to bring about vast changes to the automotive sector, as it has shown great potential to assist drivers to avoid car accidents on roads by assisting them in many different conditions. ADAS provides drivers with essential information, monitors the driver or environment, and provides guidelines to driver, which aims to improve drivers' driving experience and drivers' safety on the road.

In this paper, we propose a real-time driver alert system which aims 1) to detect the driver's eye, 2) to calculate a PERCLOS (percentage eye closure) ratio, and 3) to evaluate the driver's eyes condition, using a Raspberry Pi camera mounted on a Raspberry Pi 3. When the PERCLOS ratio exceeds a threshold, an alert system installed, in the form of a medium loud beeping sound, is switched-on. The real-time driver alert system consists of (1) a drowsiness warning system in the Driver Drowsiness Monitoring System (DDMS) module, and, (2) a Raspberry Pi development board, a camera, two LEDs, and a buzzer in the Raspberry Pi Deployment (RPD) module. The DDMS module focus on studying the driver's condition in a tri-step method: (1) detecting driver's face, (2) detecting eyes region using facial landmarks, and (3) calculating the rate of eyes closure, i.e. drowsiness level, in order to monitor the driver's condition. The RPD module focuses on (1) embedding the DDMS system into the Raspberry Pi, mounted with a Raspberry Pi camera with a buzzer, to capture the driver's condition, and (2) mounting the system in the car, to alert the driver by providing a medium loud beeping sound.

2. METHODOLOGY

In this work, we proposed a vision-based real-time driver alert system, aimed mainly to monitor the driver's condition. The alert systems consist of a (1) DDMS module, and a (2) Raspberry Pi Deployment (RPD) module. Fig. 1 shows the proposed DDMS framework. The proposed framework aims to detect a driver's eyes, to measure the driver's inattentiveness through a facial attention angle, i.e. based on the frontal face angle, and a drowsiness threshold, i.e. based on the eyes aspect ratio, to provide an alert system to the driver, in real-time. Using the personal computer or laptop may not be a suitable implementation platform for deploying a real-time alert system [9, 10]. Therefore, in this paper, a second module, being the RPD module, is proposed. The RPD module focuses on embedding our earlier proposed DDMS module in an embedded platform [11] that uses an ARM Cortex-A53 processor and 1GB of RAM [12]. The platform could be optimized to allow the system to provide notification in real-time.

3. DDMS MODULE

The framework, shown in Fig. 2, firstly obtains a video stream from the camera installed in the Raspberry Pi 3. The video frames, obtained from the threaded video file stream in H.264 raw format, is

(a) (b) (c)

Fig.3: *Video Acquisition.*

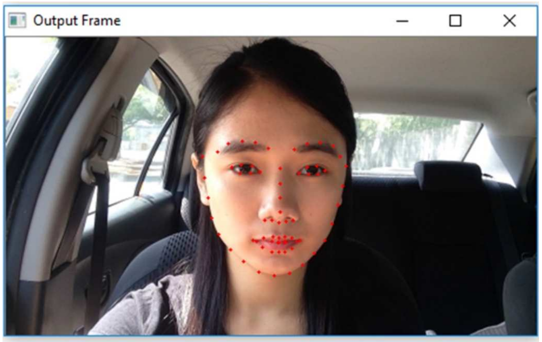
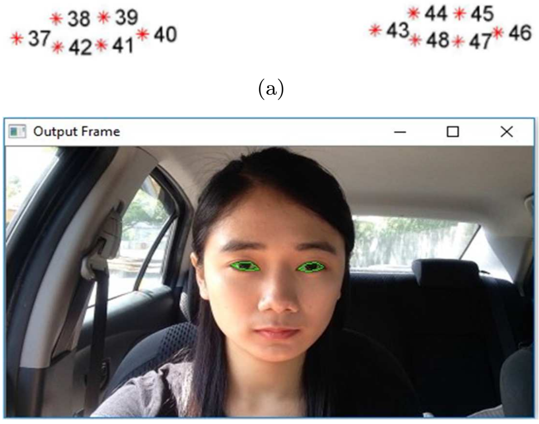


Fig.4: *Face Detection.*

being pre-processed. Later, we resized it to 640x480, and we convert it to grayscale, suitable for real-time processing. Thereafter, the facial landmarks such as eyebrows, nose, mouth, eyes and jawline were obtained. The eyes were then localized to calculate the PERCLOS, which is based on the Eye Aspect Ratio (EAR). The EAR was obtained once during the calibration period, at the system set-up time, which was used as the drowsiness threshold in the system. Later, during the journey, the EAR for both the driver's eyes will be calculated, for every frame, to determine the driver's condition. Also, the head pose is obtained to determine if the driver is attentive or inattentive. The drowsiness detection and the inattentive detection works in parallel and an alarm will be triggered when either (1) the EAR exceeds the drowsiness threshold, or (2) the driver head pose is not looking ahead.

3.1 Step 1 : Video Acquisition and Pre-processing

A 5MP Raspberry Pi camera will be mounted on a Raspberry Pi 3 model B and be placed beside the rear view mirror in the car. The Raspberry Pi unit, enclosed inside the Raspberry Pi casing, is being held by the adjustable GPS holder - see Fig. 3(a). The adjustable GPS holder, shown in in Fig. 3(b) allows the driver to adjust the position of the enclosed unit, and is based on the sitting position (and height) of the driver accordingly, whether for a left seat driver or a right seat driver. The installed position of the camera module should avoid blocking the sight of the driver to ensure it could capture the driver's face fully. The Raspberry Pi module is powered-up by the car adapter. When the system starts, it will collect the frontal scene as the HD stream- see Fig. 3(c). Later, this video stream will be resized to 640×480 pixels with 30 frames per seconds encoded in H264 format, in grayscale.



(a)

(b)

Fig.5: *Eyes Detection (a) facial landmark coordinates for eye, and (b) localized eye using convex hull.*

3.2 Step 2 : Face Detection

In this step, the resultant image in *Step 1* is processed to obtain 68 facial landmarks, using the dlib library. This library detects the facial features from the input image and then localizes all 68 coordinates on the face to determine accurately all facial features, such as eyebrows, nose, mouth, eyes and jawline. Fig. 4 shows the result of detecting all 68 facial landmark coordinates on the face. The eye region is then used to reliably estimate the condition of the eye.

3.3 Step 3 : Eyes Detection

The localization of the eye region is being determined from the 68 facial landmark coordinates mapped [12], i.e. the right eye can be accessed through landmark points 36 to 42, and the left eye can be accessed through landmark points 42 to 48 - see Fig. 5(a). The localized eyes will be marked using the convex hull function available in the OpenCV library, as shown in Fig. 5(b).

Later, the head pose is estimated using the

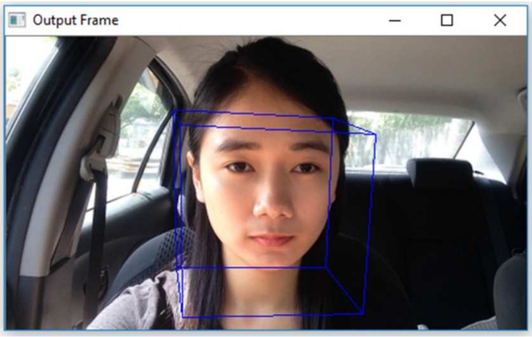


Fig.6: *Head Position Detection.*

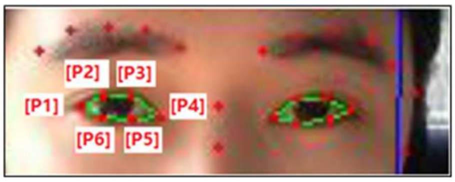


Fig.7: *EAR calculation.*

solvePnP function available in the OpenCV library. The world coordinates calculated, which is the 3D coordinates of various face points, is used to detect the head position. This pose is then re-projected from the 3D coordinate points to the world coordinate axis - shown in Fig. 6.

3.4 Step 4 : Calibration

In this step, the normal EAR is estimated [14]. This value is obtained to allow the system to measure different driver's eyes size, i.e. to allow the system to detect the normal eyes size. Thereafter, this EAR will be used as the drowsiness threshold for each driver. In this step, at first, the driver is required to stare at the camera for 5 seconds. Each eye will be detected and will be represented by all six x, y coordinates, for each eye, as in *Step 3*, shown in Fig. 7, being the P1, P2, P3, P4, P5, and P6 points. The EAR equation, shown in equation (1), computes the distance between vertical eye landmarks with the distance between horizontal eye landmarks. The EAR will normally have a constant value when the eye is open while it will drop sharply when the eye is closed. Fig. 7 shows the output for EAR from each frame from a camera with 30 fps. In preliminary experiments, the EAR for 150 frames, collected from the 5 seconds calibration phase, will be used. An EAR threshold value, being 0.24, was calculated - see Fig. 8. An alarm will only be triggered when the driver's EAR is below the threshold, named the drowsiness threshold.

$$EAR = \frac{\|P2 - P6\| + \|P3 - P5\|}{2\|P1 - P4\|} \qquad (1)$$

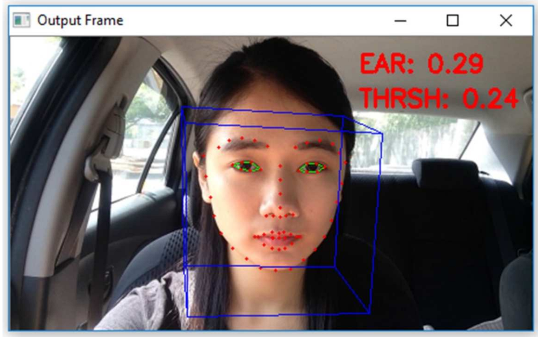


Fig.8: *Calibration process.*

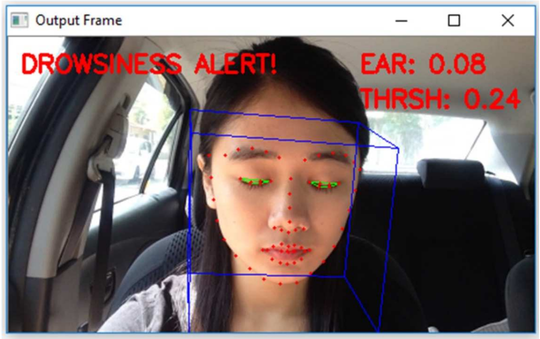


Fig.9: *Drowsiness Detection.*

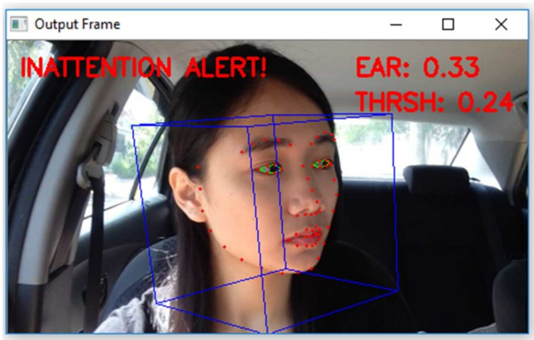


Fig.10: *Inattentive Detection.*

3.5 Step 5 : Drowsiness Detection

Any frame with EAR below the drowsiness threshold set in *Step 4*, will be recorded. If for a numbers of consecutive frames, i.e. for more than 60 frames, with EAR below the drowsiness threshold, an alarm will be triggered to alert the driver, as shown in Fig. 9. The average number of frames per second is approximately 30, and the number of frames to be considered in the drowsiness detection (for 60 frames) is equal to two seconds, which means that when the slightly closes his/her eyes for 2 seconds, an alarm will be triggered to alert the driver. The alarm will be turned off once the driver reopens his/her eyes with an EAR above the threshold value.

In order to model the head pose in a 3D space, three Euler angles of rotation around three axes are used. The three Euler angles are roll, pitch and yaw. After the Euler angles are computed, the value x, y and z of the head position is generated. By tracking the value of the y axis, we are able to determine if the driver is turning their head to the left or to the right - shown in Fig. 10.

3.6 Step 6 : Inattention Alert

In this final step, an alert will be issued when the system detects either 1) the driver is getting drowsy, or 2) the driver is not paying attention while driving. A buzzer, which is enclosed inside the Raspberry Pi casing, is connected to the GPIO of Raspberry Pi GPIO pin 33 (positive, red) and pin 39 (negative, black), to provide a medium loud beeping sound. This sound is used to alert the driver if he/she is in drowsy condition or inattentive condition.

4. RPD MODULE

The RPD module embeds the DDMS module in the Raspberry Pi platform. The RPD module consist of (1) a Raspberry Pi 3 Model B with the DDMS module embedded in it, (2) a piezoelectric buzzer, (3) a Raspberry Pi camera, (4) a micro USB power supply (2.1A), (5) a micro SD card with NOOBS (16GB/32GB), and a GPS holder, mounted behind the back mirror facing the driver; this is a low cost (about USD80) and a practical solution.

The DDMS module is embedded into the Raspberry Pi development board loaded with Raspbian-OS and attached to a 5MP Raspberry Pi camera. This camera is connected to the CSI camera connector, which is situated between the Ethernet and HDMI ports with silver connectors facing the HDMI port. The camera is fixed onto the Raspberry Pi casing facing outside of the Raspberry Pi in order to make sure the camera module is stably monitoring the driver. The buzzer, connected to Raspberry Pi GPIO pin 33 (positive, red) and pin 39 (negative, black), could emit a medium loud beeping sound to alert a drowsy driver. Finally, the micro USB power supply is plugged into the Raspberry Pi and is mounted onto the GPS holder. The whole unit is then mounted behind the back mirror facing the driver. Then, the device is being powered-up by plugging the micro USB power supply adapter into the cigarette lighter socket in the car, connecting the Raspberry Pi to the power supply.

The system requires about 35-40 seconds to boot. A beeping sound shall be emitted to indicate the system is running. Then, the calibration process begins, whereby, the driver is required to stare at the camera for five seconds, in order to obtain the drowsiness threshold. After the calibration process, the system will again emit another beeping sound to indicate that the system is ready for operation. When driver's eyes closed for two or more seconds, the system will trigger an alarm until the driver's eyes re-open again. At the same time, the system will also trigger an alarm when the driver head pose is not looking straight/forward. The system will keep running until the system fails to obtain power supply. The setting up process for a left seat driver and a right seat driver would be the same.

SetI (1) *SetI (2)* *SetI (3)* *SetI (4)* *SetI (5)*

Fig.11: *SetI : Scenario of Driver.*

5. EXPERIMENTAL RESULT

In this section, two sets of experiments were carried out. Experiment *SetI* is used to test the driver alert system in different scenarios with (1) the same driver, (2) the same environment, (3) the same lighting condition, and (4) the same vehicle. Five sets of scenarios were tested: (1) normal, (2) driving wearing eyeglasses, (3) driving wearing sunglasses, (4) driving with a cap/hat, and (5) driving with a mask. Experiment *SetII* tests the prototype with six different people. In this set-up, drivers are categorized into two main groups: (1) driver not wearing eyeglasses, and (2) driver wearing eyeglasses. In the experiment *SetII*, we invited the candidates (1) without wearing the eyeglasses to drive on road using the prototype system, and (2) wearing eyeglasses to drive using the prototype system. The drivers drove the same vehicle with the same lighting conditions to ensure the results' consistency.

5.1 Set I : Robustness Test Experimental Results

The accuracy of the drowsiness detection and inattentive detection were collected and shown in Table 1. For each scenario in *SetI* - see Fig. 11, the perfor-

Table 1: *Drowsiness detection and inattentiveness alert experimental results.*

Scenario	EAR threshold	Expected drowsy count	Detected drowsy count	Expected inattentive count	Detected inattentive count
SetI (1)	0.24	2	2	2	2
SetI (2)	0.29	2	1	2	1
SetI (3)	0.18	2	0	2	1
SetI (4)	0.22	2	1	2	0
SetI (5)	-0.05	2	N/A	2	N/A

Table 2: *Five different scenarios for face detection.*

Scenario	Video Length	Total Frame	Frame detected with face	Accuracy of frame detected with face (%)
SetI (1)	56s	1680	1401	83.39%
SetI (2)	54s	1620	1260	77.77%
SetI (3)	49s	1470	427	29.05%
SetI (4)	55s	1650	1075	65.15%
SetI (5)	54s	1620	12	0.74%

mance of the driver alert system was reported.

Scenario *SetI(1)* experimental results for both drowsiness and inattentive detection were as expected. The result in the scenario *SetI(2)*, in which the driver was wearing glasses, obtained the actual experimental results achieved 1, out of the 2 expected results. Then, the experimental result of scenario *SetI(3)* shows the system is not able to detect the drowsiness when the driver is wearing sunglasses but it performed better on inattentive behavior detection. However, the experimental results of the scenario *SetI(4)* has the opposite performance compared to the scenario *SetI(3)* where the drowsiness detection performed better than the inattentive behavior detection. The result for scenario *SetI(5)* has the most inaccurate threshold result. The drowsiness threshold obtained a value of -0.05 which is due to the failure of detecting facial landmarks during the face detection step, resulting in the error at the calibration phase.

We further analyse the face detection results which could influence the detection accuracy. Table 2 shows different video length, total frames obtained, total frames with face detected, and the accuracy obtained among five different scenarios. The experimental results in Table 2 show that *SetI(1)* scenario has the highest percentage of frames detected with driver's face, i.e. approximately 83%, and followed by scenario *SetI(2)*, with 77% succesful rate. In scenario *SetI(3)* approximately 29% of frames were detected with driver's face, while scenario *SetI(4)* obtained a better result, with 65.15% frames detected with face. In scenario *SetI(5)*, only 12 out of 1620 frames were detected with driver's face, i.e. approximately 0.74%.

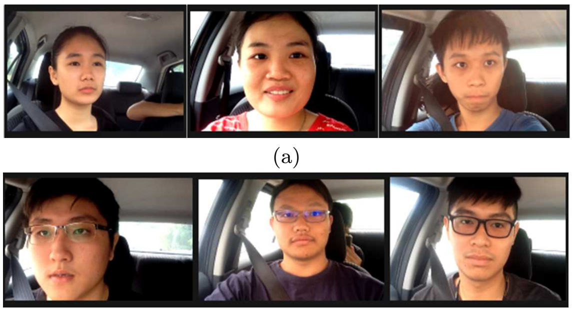
From the experimental results above, we can conclude that the designed framework might not be able to perform well for the following scenario: (1) the driver is wearing sunglasses or (2) the driver is wearing a mask.

5.2 Set II : Robustness Test Experimental Results

The experimental results of different drowsiness thresholds were recorded. The results for the drowsiness detection and inattentive detection obtained were reported in Table 4–6. Table 3 describes the results for the driver alert system concerning the drowsiness and inattentive detection results: (1) a true positive result indicates that the alarm will be triggered when the drowsiness or inattentive condition is detected, (2) a false positive indicates the alarm will be triggered when there is no sign of drowsiness or inattentive condition detected, (3) a true negative indicates that the alarm is off when the driver did not exhibit any drowsiness or inattention condition detected, and (4) a false negative result indicates that the alarm is not triggered even when there is a sign of drowsiness or inattentive condition detected.

In this section, the performance of the driver alert system was reported. The experimental results for the drowsiness detection system for drivers with and without eyeglasses were carried out in experiment *SetII* - see all scenarios in Fig. 12. The average accuracy for the drowsiness detection in drivers without eyeglasses achieved an astounding 85.86%, which is higher than the average accuracy for drivers with eyeglasses - see in Table 4.

The average accuracy of the inattentive detection



(a)

(b)

Fig.12: *SetII : Scenario of Driver for (a) not wearing eyeglasses- Set II(1) SetII(2) SetII(3) [left to right] (b) wearing eyeglasses - Set II(4) SetII(5) SetII(6) [left to right]*

Table 3: *Driver alert system description.*

Scenario		Condition	
		Drowsy / Inattentive	Not Drowsy/ Attentive
Test	Alarm On	TP	FP
	Alarm Off	FN	TN

Table 4: *Drowsiness detection experimental results.*

SetII	Dataset	TP	TN	FP	FN	Accuracy
Without eyeglasses	SetII (1)	469	1033	129	125	85.54%
		Total: 1502		Total: 254		
	SetII (2)	168	854	166	12	85.17%
		Total:1022		Total: 178		
	SetII (3)	180	972	54	120	86.88%
		Total: 1152		Total: 174		
Average drowsiness detection accuracy:						**85.86%**
With eyeglasses	SetII (4)	108	1060	264	269	68.67%
		Total: 1168		Total: 533		
	SetII (5)	210	1011	325	192	70.01%
		Total: 1221		Total: 517		
	SetII (6)	246	1034	236	210	74.16%
		Total:1280		Total: 446		
Average drowsiness detection accuracy:						**70.95%**

of the drivers without eyeglasses was about the same with the average accuracy of drivers with eyeglasses, at around 73% - shown in Table 5.

Table 6 lists the drowsiness threshold value for each of the candidates from the second dataset. The drowsiness threshold values of the candidates without eyeglasses is significantly higher compared to the drowsiness threshold values of the candidates with eyeglasses. From these results, false negative detection could be due to the threshold value for the eye size of the candidate (with eyeglasses) due to the reflection of lights on the driver's eyeglasses. Inaccurate collection of driver eye size would affect the system

Table 5: *Inattentive detection experimental results.*

SetII	Total Frame	TP	TN	FP	FN	Accuracy
Without eyeglasses	SetII (1)	182	1151	186	237	75.91%
		Total: 1333		Total: 423		
	SetII (2)	150	701	190	159	70.92%
		Total:851		Total: 349		
	SetII (3)	278	712	208	128	74.66%
		Total:990		Total: 336		
Average inattentive detection accuracy:						**73.83%**
With eyeglasses	SetII (4)	174	1161	48	269	66.78%
		Total: 1136		Total: 366		
	SetII (5)	174	1161	48	269	66.78%
		Total:1386		Total:358		
	SetII (6)	253	1016	239	210	73.52%
		Total:1269		Total: 457		
Average inattentive detection accuracy:						**73.26%**

Table 6: *Drowsiness threshold results.*

SetII	Dataset	Drowsiness threshold	Average threshold
Without eyeglasses	SetII (1)	0.33	0.35
	SetII (2)	0.34	
	SetII (3)	0.38	
With Eyeglasses	SetII (4)	0.17	0.22
	SetII (5)	0.26	
	SetII (6)	0.22	
Average drowsiness threshold value:			0.28

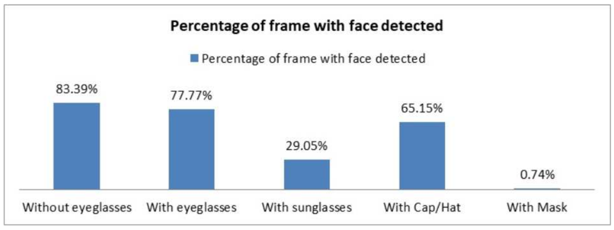


Fig.13: *Chart of percentage of frame with face detected.*

to produce inaccurate drowsiness threshold value.

6. DISCUSSION AND FUTURE WORKS

Fig. 13 shows the precentage of frames detected with driver's face from the five different scenarios carried out in experiment SetI. Scenario SetI (1), without the eyeglasses, has the highest percentage of frames detected (83.39%) while the last scenario with the driver wearing a mask while driving the car has only 0.74% of frames detected - see Fig. 14.

The data shown in Fig. 15 were collected and generated from the results obtained from the experiment SetI. The first category of experiment was carried out by the same driver in the same environment driving the same vehicle for different scenarios. The re-

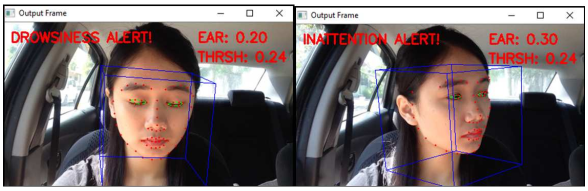


Fig.14: *Detection results (left) drowsiness alert and (right) inattention alert.*

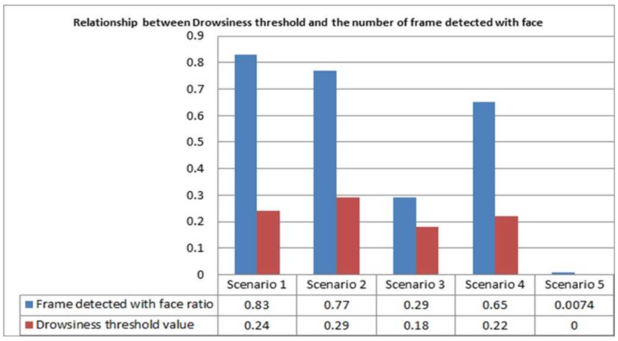


Fig.15: *Relationship between drowsiness threshold and number of frames detected with face .*

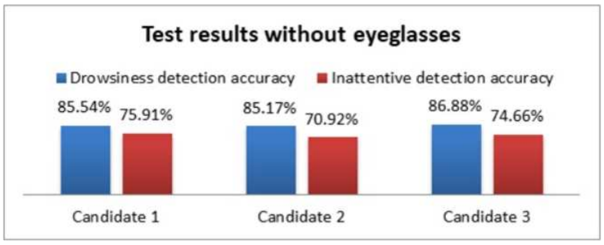


Fig.16: *Test results without eyeglasses: percentage of frames detected (y-axis) with different candidate's face detected (x-axis).*

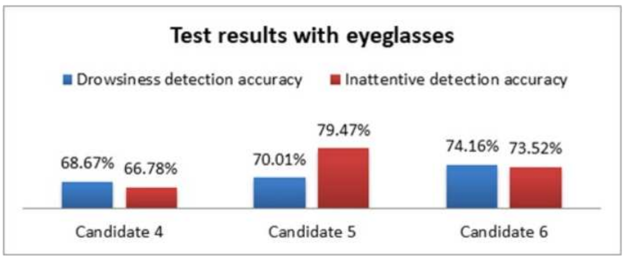


Fig.17: *Test results with eyeglasses: percentage of frames detected (y-axis) with different candidate's face detected (x-axis).*

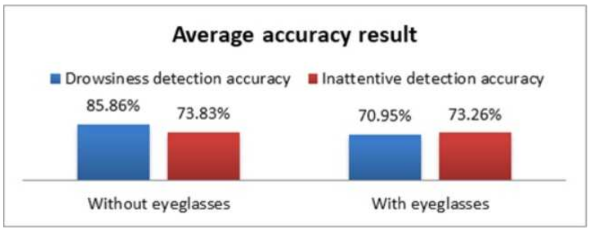


Fig.18: *Average accuracy results: percentage of frames detected (y-axis) for different conditions (x-axis).*

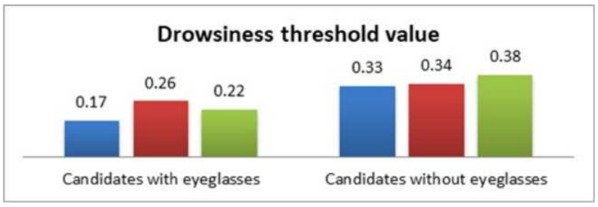


Fig.19: *Drowsiness threshold values (x-axis) for different conditions (x-axis).*

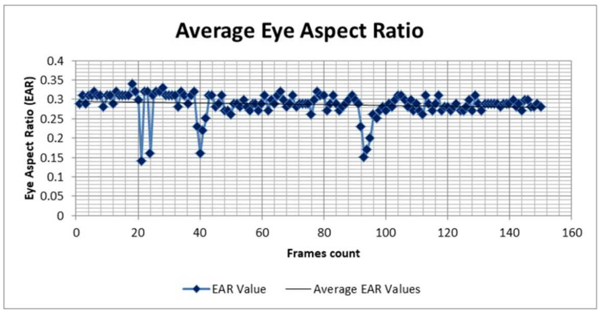


Fig.20: *Average Eye Aspect Ratio (EAR) calculated during calibration process.*

sult above shows that the drowsiness threshold values were different from each scenario even though all the scenarios were carried out by same driver with the same eyes size. These results show that a relationship exists between the percentage of frames detected and the drowsiness threshold value. Percentage of frames detected values were obtained from the number frames detected over the total frames of each scenario. While the drowsiness threshold values were obtained after the calibration process for each of the scenarios above. The chart above shows that the drowsiness threshold was affected by the percentage of frames detected. The drowsiness threshold value is lower in value when the percentage of framed detected was low. However, when the percentage of framed detected is higher, the value of drowsiness threshold was higher.

Fig. 16 shows the results obtained from experiment *SetII*, for drivers driving without wearing any eyeglasses. Drowsiness detection and inattentive detection were carried out in this experiment and the accuracy of both detections is calculated. The experimental results above indicate that drowsiness detection has higher accuracy compared to the accuracy of inattentive detection for all three candidates who were driving with the prototype system in the same lighting conditions, same environment, and same vehicle. Thus, it can be concluded that the performance of drowsiness detection is better than the performance of inattentive detection in the proposed framework.

Fig. 17 shows the results obtained from experiment *SetII*, for drivers driving with eyeglasses. The experimental results obviously show that the overall percentage of accuracy for both detections was lower compare to the results without eyeglasses. As observed, the drowsiness detection accuracy was significantly lower compared to the result of inattentive detection for all three candidates. However, the result of inattentive detection did not have big differences compared to the results without the eyeglasses. Therefore, it can be concluded that the presence of eyeglasses will affect the result of drowsiness detection.

Fig. 18 shows the experimental result obtained from both drowsiness detection and also the inattentive detection for different sets of tests. The average accuracy of drowsiness detections based on the two different sets of tests shows a significant difference. The average accuracy of drowsiness detection in drivers driving without eyeglasses is approximately 85%, which is much higher compared to the average accuracy of the drivers driving with the eyeglasses which is approximately 70%. Besides that, average accuracy of inattentive detection for both test sets was approximately at the same percentage which is 73%. From the results above, it can be concluded that eyeglasses will affect the performance of drowsiness detection but did not have much affect on inattentive detection. Drivers are advised not to wear any anti-reflective/anti-glare eyeglasses while driving. Lights reflecting on the eyeglasses will reduce the accuracy in locating the driver's eyes, which leads to low performance in the drowsiness detection.

Fig. 19 shows the results obtained in the experiment *SetII*. The results in this figure indicate the drowsiness threshold value for each of the candidates from both test scenarios in experiment *SetII*. The drowsiness threshold values of the candidates without eyeglasses is significantly higher compared to the drowsiness threshold values of the candidates with eyeglasses. From the result above, it can be concluded that the prototype system might obtain an inaccurate value for the eye size of the candidate with eyeglasses due to the reflection of lights on the driver's eyeglasses. Inaccurate location of driver's eye size will cause the system to generate an inaccurate drowsiness threshold value.

Fig. 20 provides the information on the average EAR calculated during the calibration process for the driver in the first experiment. During the calibration process, the EAR values were collected for a total of 150 frames. As can be seen from the graph above, the EAR value was approximately constant until frame 20, 40 and 90. The ratio dropped sharply indicating that the eyes were closed during the frames. Hence the average Eye Aspect Ratio (EAR) value of this particular driver's eyes was computed with value 0.28 and the drowsiness threshold value generated as 0.24 in this case. The prototype system is then used to monitor the driver state such as drowsiness or inattentive. When the prototype system detects that the driver is in a drowsy state, the prototype system will alert the driver with a medium loud beeping sound. At the same time, the prototype system can also detect if the driver is not looking ahead or in inattentive driving state. The prototype system will also alert the driver with a loud beeping sound when the driver is in inattentive driving state such as looking left or right outside the car.

However, there are also some weaknesses in the prototype system. One of the issues of the system is that it is not able to function at night. The system is not able to detect the face and eyes of the driver at night. Hence it is not able to determine if the driver is getting drowsy. By attaching a flash light beside the camera, it will distract the driver while driving. The flash light will also cause the driver to bot see clearly during the night which is very dangerous while driving during the night. The second issue is that if the driver missed the calibration step by staring at the camera for less than 5 seconds after the system booted up, the driver might encounter many errors when operating the system. Once the system did not detect any eyes from the frame, the threshold will set to zero. When the threshold is set to zero, it is impossible to detect when a driver is drowsy which will cause serious consequences. In addition, if the driver did not mount the prototype system properly, by positioning the camera in the right position on the car, allowing the camera to capture the driver's face fully, the system might not able to perform well. If the system is mounted too high or too low in the car, the camera might lose the detection of the face which will leads to drowsiness and inattentive behaviors detection not functioning properly. In another case, if the system is mounted too near to the face, the system might not be able to capture the whole face of the driver leading to the same false detection results discussed earlier. Further, we compared the proposed real-time driver alert-system with other works - see Table 7.

For the future enhancement and development, a night vision camera will be implemented. It is to allow the system to assist drivers at night to ensure the drivers could stay alert. Besides that, future works include studying different states to measure the driver attentiveness. These states could include more de-

Table 7: *Comparison with other works.*

Method	Driver Drowsiness Module	Deployment Module	Real-time Capabilities
Eskandarian [4]	Yes (steering and driving)	No (Simulator)	No
Musale [3]	Yes (based on eye)	Not reported (live videos)	No
You [6]	Yes (including, driving speed, turn and lane trajectory)	Yes (smartphone)	Yes (not mentioned specification/calibration)
Proposed System	Yes (EAR and headpose)	Yes (low-cost, easy implemented, Raspberry Pi)	Yes (up-to 30 fps)

tails such as yawning, head nodding and/or other head pose. Even though driver drowsiness detection systems have existed for a long time, driver drowsiness monitoring systems still have much room for improvements, such as real-time implementation of the system. This paper proposed a low cost and practical real-time driver monitoring and alert system in ADAS perspective to decrease the car accident rate due to the drowsiness of the driver during driving the car.

References

[1] "Global Status Report On Road Safety," 2015. [Online] Gevena: The World Health Organization. Available: `http://www.who.int/violence_injury_prevention/road_safety_status/2015/GSRRS2015_Summary_EN_final2.pdf` [Accessed 26 Nov. 2017].

[2] B. Jansen, "Report: Drowsy driving is a sleeper threat in crashes," [Online] USA TODAY. Available: `https://www.usatoday.com/story/news/nation/2016/08/07/report-drowsy-driving-sleeper-threat-crashes/88300112/` [Accessed 26 Nov. 2017].

[3] T. Musale and B.Pansambal, "Real Time Driver Drowsiness Detection system using Image processing," *Research in Engineering Application & Management (IJREAM)*, vol.2, pp.35-38, 2016.

[4] A. Eskandarian and A. Mortazavi, "Evaluation of a Smart Algorithm for Commercial Vehicle Driver Drowsiness Detection," in *2007 IEEE Intelligent Vehicles Symposium*, Istanbul, pp. 553-559, 2007.

[5] S. Vitabile, A. D. Paola and F. Sorbello, "Bright Pupil Detection in an Embedded, Real-Time Drowsiness Monitoring System," in *2010 24th IEEE International Conference on Advanced Information Networking and Applications*, Perth, WA, 2010, pp. 661-668.

[6] C. You, N.D. Lane, F. Chen, R. Wang, Z. Chen, T.J. Bao, M. Montesde-Oca, Y. Cheng, M. Lin and L. Torresani, "Carsafe app: Alerting drowsy and distracted drivers using dual cameras on smartphones," in *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, pp. 13-26, 2013.

[7] J. Wang, D. Liu, K. Zhang, X. Chen and J. Kim, "A novel real-time service architecture based on driver state detecting for improving road safety," in *2015 IEEE International Conference on Consumer Electronics - Taiwan*, Taipei, pp. 53-54, 2015.

[8] Tejasweeni Musale and B.H. Pansambal, "Driver Drowsiness Detectiontechnique Using Raspberry Pi," *International Journal of Development*, Research Vol. 07, Issue, 02, pp.11499-11503, Feb 2017

[9] W. Chang, L. Chen and Y. Chiou, "Design and Implementation of a Drowsiness-Fatigue-Detection System Based on Wearable Smart Glasses to Increase Road Safety," in *IEEE Transactions on Consumer Electronics*, vol. 64, no. 4, pp. 461-469, Nov 2018.

[10] Y. Kortli et al., "Efficient Implementation of a Real-Time Lane Departure Warning System," in *IEEE International Image Processing Applications and System Conference 2016 (IEEE IPAS2016)*, Hammamet, Tunisia, 5-7 Nov 2016.

[11] W. Chee, P.Y. Lau, S. Park, "Real-time Lane Keeping Assistant System on Raspberry Pi," in *IEIE Transactions on Smart Processing and Computing*, vol. 6, no. 6, Dec 2017.

[12] Pascal Francis-Mezger and Vincent M. Weaver. 2018. A raspberry pi operating system for exploring advanced memory system concepts. in *Proceedings of the International Symposium on Memory Systems (MEMSYS '18)*. ACM, New York, NY, USA, 354-364.

[13] C. Sagonas, E. Antonakos, G. Tzimiropoulos, S. Zafeiriou and M. Pantic, "300 Faces In-The-Wild Challenge: database and results," *Image and Vision Computing*, vol.47, pp 3-18, 2016.

[14] T. Soukupová and J. Cech, Eye-Blink Detection Using Facial Landmarks, 2016.

Jie Yi Wong received her Bachelor of Information Systems from Universiti Tunku Abdul Rahman, Malaysia in 2019. Currently, she is working as a system developer in TNG Digital Sdn Bhd. Her research interests include Internet of Things, machine learning, and multimedia signal processing.

Phooi Yee Lau received her BCompSc. from Universiti Teknologi Malaysia, Malaysia in 1996, MCompSc. from Universiti Malaya, Malaysia in 2002, and Ph.D. in Engineering from Keio University, Japan in 2006. She is currently an Associate Professor at Universiti Tunku Abdul Rahman. Her current research interests include intelligent system, media communication, and video delivery.