2016 12th International Conference on Signal-Image Technology & Internet-Based Systems

Driver Drowsiness Detection using Eye-Closeness Detection

Oraan Khunpisuth, Taweechai Chotchinasri, Varakorn Koschakosai
Mahidol University International College,
Department of Computer Engineering,
Faculty of Engineering, Mahidol University, Nakhon Pathom, Thailand
{orann, pc.imping, kornnufc5}@gmail.com

Narit Hnoohom
Image, Information and Intelligence Laboratory,
Department of Computer Engineering,
Faculty of Engineering, Mahidol University, Nakhon Pathom, Thailand
narit.hno@mahidol.ac.th

***Abstract*—The purpose of this paper was to devise a way to alert drowsy drivers in the act of driving. One of the causes of car accidents comes from drowsiness of the driver. Therefore, this study attempted to address the issue by creating an experiment in order to calculate the level of drowsiness. A requirement for this paper was the utilisation of a Raspberry Pi Camera and Raspberry Pi 3 module, which were able to calculate the level of drowsiness in drivers. The frequency of head tilting and blinking of the eyes was used to determine whether or not a driver felt drowsy. With an evaluation on ten volunteers, the accuracy of face and eye detection was up to 99.59 percent.**

Keywords-Drowsiness; Face Detection; Eye Detection; Geometric Rotation; Region of Interest

I. INTRODUCTION

Car accidents are a major problem for society, with statistics for the rate of injury or death as the result of a car accident rising. There is a fatal car accident about every 25 seconds [1]. Car accidents annually cost USD $518 billion worldwide. In developing countries such as Thailand, the roads are dangerous. Car accidents can be seen on the Thai news every day. According to the University of Michigan Institute of Technology, Thailand's roads were the second most dangerous in the world in 2013 [2]. Solutions are needed.

There are numerous non-driver related causes of car accidents including road conditions, the weather and the mechanical performance of a car. However, a significant number of car accidents are caused by driver error. Driver error includes drunkenness, fatigue, and drowsiness. Many factors can affect a driver's ability to control a motor vehicle, such as natural reflexes, recognition and perception. The diminishing of these factors can eventually reduce a driver's vigilance level [3].

Statistically, drowsiness by drivers results in an estimated 1,550 deaths, 71,000 injuries, and $12.5 billion in monetary losses [4]. As mentioned above, the situation in Thailand is even worse compared to the global average. Car accidents are a daily occurrence on the news in Thailand. Throughout the centuries, there have been numerous "raising awareness" campaigns about drowsiness and drunken driving in Thailand. However, they have been ineffective for the most part. Auto accidents not only affect the drowsy drivers, but also any potential victims.

During the years, several real-time face and eye detection techniques have been developed [5-7] for monitoring the driver drowsiness. This paper aims to evaluate drivers' specific activities to determine their level of drowsiness. The authors have implemented a Raspberry Pi device programmed with an innovative algorithm. This algorithm allows a Raspberry Pi camera module to be able to detect the face and eyes, which are considered two of the most significant activities when driving that can serve as factors to gauge the drowsiness level of drivers. Since precise detection was achieved using the interested values, the authors expect that the device can accurately gauge driver drowsiness.

Volkswagen has developed a driver fatigue detection system, which can automatically monitor an operator's driving characteristics [3]. Basically, the system will recommend that the driver take a break if it detects possible fatigue. This system is configured to continually evaluate driver's fatigue based on various factors such as driving speed, steering wheel movement, the driver's head position, and the driver's eyes. If the system concludes that the driver is exhausted, the driver will be warned by information in the multi-function display together with an acoustic signal. The system will repeatedly warn the driver for fifteen minutes if the car has not stopped. However, the driver remains in charge of the vehicle.

Figure 1. Warning Message via Display Monitor in a VW Car [8]

After extensive research to define fatigue, Volkswagen began to implement its device in cars. Volkswagen's driver fatigue detection includes two main parts, including a video sensor and prediction algorithm. A. The video sensor is used to provide an image of the driver in all lighting conditions. This includes image processing software as an element. Image processing is responsible for identifying certain important parameters such as eyelid movement, head position and gaze position. A prediction algorithm is used to calculate the fatigue level of the driver. Its calculation will be based on the position of the driver's head at certain intervals, eyelid closure, and blink rate. This basis provided the concept for the authors to develop Driver Drowsiness Detection, which is used to detect or not a driver is drowsy. This paper introduces an alerting process for when the driver fall asleep. It calculates the level of drowsiness depending on the frequency of head tilting and eye blinking rate. If the level exceeds the limit from a threshold, this study's embedded device will alert the driver.

As shown through the innovation in this paper, the solution used to check for whether or not a driver is drowsy possesses good accuracy. On the other hand, the face and eye detection algorithm provided in OpenCV library is not as accurate. Some parts of the algorithm cause false positive readings. Also, it is unable to detect the tilting head of a driver. The Haar Cascade Classifier is trained to detect only the front view of people's faces. This paper addresses these problems by using different methods such as region of interest, geometric rotation and template matching. As such, this research provides better results than the document.

There are some limitations in this paper. The first limitation deals with lighting conditions. Raspberry Pi's camera is used to detect the faces and eyes of drivers. The paper experimented in a location with constant light control, so the camera could trace the face and eyes efficiently and accurately. Another limitation relates to eyeglasses. The face detection method, which is the Haar Cascade Classifier, will not work efficiently when the driver wears eyeglasses. Lastly, drivers with very dark skin tones are not able to be detected with accuracy [9]. The Haar Cascade Classifier is trained to recognise persons with average or neutral skin colour. Moreover, an expectation of this paper is the suggestion of additional solutions and techniques, which the authors sincerely hope can help improve facial detection algorithms. Therefore, the authors gratefully allowed other programmers to solve the limitations in order to improve the efficiency of this work and any study related to this field.

II. METHODOLOGY

A. Overview Design

The main concept of Driver Drowsiness Detection is to capture a driver's face from a camera and be able to accurately calculate the level of drowsiness in drivers with real-time processing. To achieve these requirements, proper materials have to be selected. For the base computer, the research team selected Raspberry Pi 3 Model B. The Raspberry Pi camera was the visual device of choice. Moreover, hardware works great with well-designed software. As such, software is the most important part in any designed device. For video and photo manipulation, the research team chose OpenCV library for its completeness and convenience.

B. Face Detection

Driver Drowsiness Detection requires a video sensor to detect the faces of drivers. A Pi Camera was employed in this capacity. Also, it is processed by Raspberry Pi 3. The authors detected the drowsiness level of drivers by checking for head tilting and eye blinking rate. Therefore, there are several methods that are applied in this paper.

1) *Haar Cascade Classifier*

The first tool utilised to detect a driver''s face is the Haar Cascade Classifier. It is one of the few object detection methods with the ability to detect faces. Paul Viola and Michael Jones invented this method [10]. It is trained with thousands of faces in different light conditions [11-12]. Also, the method efficiently detects people in a bright room. In this work, the Haar Cascade Classifier was used to detect the face because it is considered to have a very high accuracy rate.

A percentage of face detection is about 83.09 percent based on the test cases in this study with ten volunteers, as seen in Table I. This method works with both men and women. Even when the eyes are closed, detection of the face is still possible. Ten people were tested with the Haar Cascade Classifier, as shown in Figure 2.

Figure 2. Ten volunteers tested with the Haar Cascade Classifier

2) *Region of Interest*

Region of interest (ROI) can detect a driver's face with increased accuracy. As can be seen in Figure 3, the blue rectangle is the region of interest.

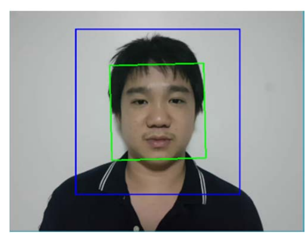

Figure 3. Region of Interest

The way to create an ROI area is to first obtain the green rectangle area from the Haar Cascade Classifier in the first frame, which includes height, width, and the points of x and y. Then, the rectangle is scaled up to create region of interest. There are several steps to calculate the ROI area.

To calculate width and height, the authors have figured out a formula for scaling up a new rectangle. There are three variables, which are width and height, from the Haar Cascade Classifier (green rectangle), multiplier and another temporary variable (so called 'temp'). The authors multiply these variables together. The value of the multiplier will be set between 1.7 and 2 based on different stages. There are two stages of the multiplier variable. The first stage is that the authors detect the face in the first frame by using the Haar Cascade Classifier. It detects the whole frame at this stage. Then, the detection area will be smaller in the next frame due to ROI. Therefore, the authors use the Haar Cascade Classifier to detect the face only in the ROI.

The second stage is when the authors use the Haar Cascade Classifier only in the ROI area that the authors get from the last frame. At this stage of the operation, the multiplier variable will be set at 1.7 in order to create a smaller ROI area. This helps reduce false positive readings from the Haar Cascade Classifier.

Another variable included in the calculation is temp. The purpose of this variable is to control the ROI's size so that it is not over enlarged when the head is bent. To illustrate, when the driver's head is tilting, the width of the face's frame will be greater than usual. This will make the size of ROI larger than the authors expected. Consequently, the authors created the temp variable to reduce the size of the ROI based on head angle. The authors use the angle of the face to make a calculation when the driver tilts his head. The formula for variable temp can be written as:

$$T = (100 - \left|\frac{FC}{2}\right|)/100 \tag{1}$$

where T is the temp value and FC is the face degree.

The value of temp will not be more than 1 due to the fact that it is created to minimize rather than to enlarge. As a result, the authors get the value of three variables, which are width, multiply and temp. The formula of output width is:

$$OW = IW \times M \times T \tag{2}$$

where OW is the output width, IW is the input width, M is the multiply value, and T is the temp value.

In addition, maintaining the rectangle shape means the height of ROI will be derived from the same formula as the output width. The second step is to calculate for points x and y. The formula to calculate the new point of x is:

$$x = \frac{OW - IW}{2} \tag{3}$$

where x means the point x from the Haar Cascade Classifier, OW means the width from the ROI, and IW means width from the Haar Cascade Classifier.

The last step is to calculate for point y. This tends to be the same method used for point x calculation. Finally, the authors will get the value of the region of interest which are width, height, the point x and y, as can be seen in Table I. In this case, the face angle is -1.7346 degrees tilting to the left side.

a) Advantages of Region of Interest

The first advantage of ROI is the application of the Haar Cascade Classifier only in the ROI area. It is obviously faster than detecting faces from an entire frame. At the beginning of face detection, the ROI area will be created after the authors use the Haar Cascade Classifier for the whole frame. Then, the authors will use the Haar Cascade Classifier in the ROI every frame after the first frame. This will reduce the size of the detection area.

TABLE I. CALCULATION OF ROI

	Calculation of Region of Interest			
	Width	***Height***	***X***	***Y***
Input Rectangle	210	210	217	116
Output Rectangle	420	420	112	11
Input Rectangle	217	217	213	113
Output Rectangle	364	364	140	40
Input Rectangle	211	211	219	118
Output Rectangle	355	355	147	46

The second advantage deals with false positive reduction. It is possible that the Haar Cascade Classifier will detect false positives in some frames. Creating the ROI will help to minimise this problem as there are fewer chances that false positives will occur in the minimised ROI. To ensure that the program will keep tracking only the face of the driver, the authors also make the Haar Cascade Classifier focus on the biggest face in the ROI. Therefore, false positives are completely eliminated because other faces in a frame will be disregarded.

C. Eyes Detection

After the authors can detect the face of the driver, the authors calculate drowsiness level of the driver based on eye blink rate. For instance, if drivers blink their eyes more frequently, the authors assume that the drivers are drowsy. Thus, it is necessary for this paper to detect eyes accurately in order to calculate for eye blink frequency. Fortunately, the Haar Cascade Classifier, which is used for face detection in this study, provides a great ratio for eye area. To equalise various face sizes, the authors resize the images by 200×200 in order to get the same eyes ratio for everyone. The area of the eyes will be further cropped in order to erase unrelated sections. Despite the fact that the Haar Cascade Classifier provides a great ratio of eye area, it is still not as constant as the authors need. As a result, the authors use template matching to ensure that the eye area can improve its efficiency.

1) Template Matching

Template Matching is a technique used to find a specific area within an image that is most similar to the input image [13]. In this case, the authors use three templates in order to check for eye area and eye blink. The library in OpenCV already provides a template matching method. The patch image that the authors use in the template matching is the area of the eyes from one member of the authors' team, as shown in Figure 4.

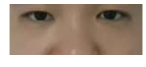

Figure 4. Eyes Template Matching

CV_TM_CCOEFF_NORMED, is selected in this paper. As the authors tested with ten test cases, this method gives better results than other template matching methods. It detects eye area very accurately. The formula is written as:

$$R(x,y) = \frac{\sum_{x',y'}(T'(x',y') \times I'(x+x',y+y'))}{\sqrt{\sum_{x',y'} T'(x',y')^2 \times \sum_{x',y'} I'(x+x',y+y')^2}} \quad (4)$$

where R is the correlation value, T is the template image, and I is sub-image of the input image.

After template matching was applied, the authors normalised the output of the matching procedure. Then the authors localised the maximum value by using minMaxLoc provided in OpenCV. According to the normalised correlation coefficient method, the higher value was the best match.

D. Eyes Closeness Detection

A combination of face detection and eye detection allows for checking of eye blink rate and eye closeness. The way the authors check whether or not the eyes of the driver are opened is to use template matching. The template image can be seen in Figure 5.

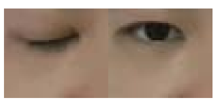

Figure 5. Templates for Closed Eye (Left) and Opened Eye (Right)

The authors only have the template for the right eye, so the solution to template matching for the left eye is to flip the template image. Therefore, it is possible to obtain four template images, which the authors use for checking in four stages, including opened left eye, opened right eye, closed left eye, and closed right eye. Moreover, the authors use a template matching method, which is a normalised correlation coefficient. minMaxLoc provides the nearest number to one. The number indicates an area that is the most similar to a template image. As illustrated in Figure 6, the value of closed eyes will be higher than for opened eyes if the driver closes his eyes.

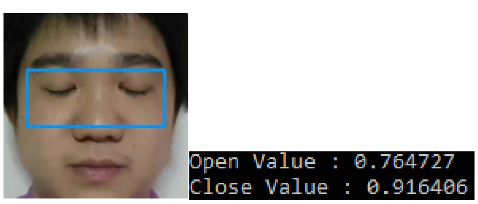

Figure 6. The Number of Closed Value is More Than Opened Value

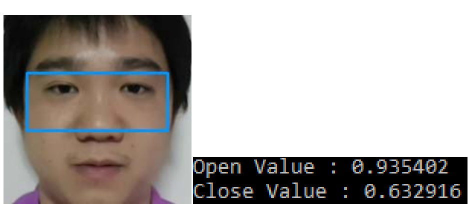

Figure 7. The Number for Opened Value is More Than Closed Value

On the other hand, the value for opened eyes will be higher than closed eyes if the driver opens his eyes, as shown in Figure 7.

E. Geometrical Rotation

The authors assume that the Haar Cascade will work operatively so long as the face is always in the front facing position. Consequently, the proposed solution is to rotate the tilting faces back into a front-facing position.

Figure 8. Tilting Head and Geometric Rotation

The first step is to make the program be able to determine whether or not the head is tilting. The authors realised that the y-axis position of two eyes could identify head tilting perfectly. As seen in Figure 8, the locations of the left and right eye will be nearly the same when the face is upright and straight. On the other hand, tilting of the head means those locations in y-coordinate will surely be changed into different positions. Therefore, as the authors can locate the eye, the authors simply make the program identify the tilting head when it occurs.

The next step is to calculate how many degrees the authors have to rotate in order to set any tilting faces straight. This step is the reason why the authors call this method the 'Geometric Rotation'. The authors use geometrical knowledge to accomplish this. After the authors obtained the x and y coordinates for both eyes, the authors were able to draw a virtual right triangle between those two eyes. The right triangle gives the ability to know how many degrees an eye is from with another. Thus, the authors apply arctan to the ratio between height and width of the triangle to obtain the desired angle.

The authors use the warpAffine function from OpenCV for rotation. The way the authors rotate the face is done frame by frame. The authors create two degree variables, which are frame degree and cumulative degree. Frame degree is the degree that the authors obtain from each frame starting from the first one. In every frame, the authors add the frame degree value to the cumulative degree, which is initially set at zero. Then, the authors reset the frame degree value to zero and rotate the image with the cumulative degree before continuing to the next frame. The next frame degree will calculate the degree from a previously rotated frame. Therefore, the value of the frame degree, which is gradually added to the cumulative degree, will be quite small. The reason for doing this is to prevent the program from potentially going horribly wrong. For instance, the effect will not be large if the calculation is incorrect and would be completely resolved in the next several frames. This means that the program still achieves high accuracy when detecting faces.

In addition, it is very crucial for the program to be able to keep tracking the current position of the face.

Therefore, the authors have to know where exactly the face is in the original frame after the face is detected from the rotated frame. There is a mathematical formula to get the new point after a rotation, which is:

$$x' = x\cos\theta - y\sin\theta \quad (5)$$
$$y' = x\sin\theta + y\cos\theta$$

where x' and y' are the coordinates of the new point, x and y are the coordinates of the original point, and θ is the angle.

It is obvious that the authors will get the original point if applying this formula in reverse. As the goal is to be able to solve these equations in reverse, the authors are supposed to solve two variable equations. Writing an algorithm to solve two variable equations is tough and consumes some computational time as it may require several loops. Therefore, the authors chose to use Cramer's rule principle to sort this out.

F. Drowsy Checking

There are two ways to check for the drowsiness level of a driver when combining two methods for face and eye detection. The first way deals with eye activity. The authors use a timer in the OpenCV library in order to account for eye blinking duration [14]. The duration of eye blinking for human beings is about 300 milliseconds, as the authors tested with ten cases. One of the ten cases is shown in Figure 9.

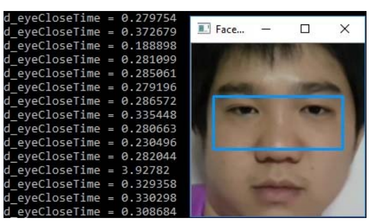

Figure 9. Normal Eyes Blinking Rate

There are several conditions that the authors use to check for a drowsy driver in terms of eye activity. The first condition is to see if the driver closes his eyes for more than two seconds, which could be considered a full closure of the eyes. On the other hand, the driver closing his eyes for less than two seconds can be translated as a blink. In this case, the authors know the frequency of eye blink rate, which the authors can use to calculate the drowsiness level of the driver.

The second way to check for a drowsy driver concerns head activities. After the authors can detect the face and eyes accurately, the authors will know the angle when the driver tilts his head. A degree value is a negative value when the driver tilts his head to the left hand side. If the value for degree is higher than the threshold that the authors set, it can be assumed that the driver is likely in danger. The threshold will be described in the next section. According to observations, the driver will not tilt his head more than thirty degrees while driving. Furthermore, if the driver is not in the camera's sight or in the frame, the driver will be considered to be in danger.

1) Threshold

The authors continue calculating for the threshold of drowsiness level. There are many conditions that lead to an increase of the drowsiness level in the driver.

Firstly, the program will determine eye blink. The driver must blink or close his eyes within three seconds each time, which is considered normal blinking. This will be counted as a blink. In addition, the driver has probably fallen asleep if closing his eyes for more than three seconds. The drowsiness level will instantly increase to one hundred, which will cause the driver to be alerted.

Secondly, the authors check for eye activity. The average rate of eye blink for human beings is approximately fifteen to twenty times in one minute [15]. However, the authors understand that some people might have different or unique blink rates. Consequently, the driver in this case will be calibrated for one minute in order to calculate the ratio for eye blink.

The formula to calculate drowsiness level is written as:

$$DL = DL + (NT - BR) \quad (6)$$

where *DL* is the drowsiness level, *NT* is the interval number, and *BR* is the number for blink rate.

If the driver blinks his eyes more than five times in the next interval, the drowsiness level will be increased. However, if the driver blinks his eyes less than five times, the drowsiness level will be decreased.

Lastly, head activities also affect the drowsiness level of the driver. The driver should not tilt his head forward more than thirty degrees. There are many conditions of a head check. At first, if a driver tilts his head for less than two seconds, the drowsiness level will increase twenty percent. If the head is tilted for more than two seconds, the drowsiness level rises forty percent. Another condition deals with the combination of eye and face activities. If the driver tilts his head for more than two seconds and the eyes are closed for more than two seconds, the drowsiness level will reach one hundred percent. As can be seen in Figure 10, the drowsiness level rises to one hundred. In addition, if the camera cannot detect the face of the driver for more than two seconds, the drowsiness level reaches one hundred percent.

The drowsiness level of the driver decreases when the face is not tilted and the eyes blink normally. Another condition that can reduce the drowsiness level relates to the driver, who should drive properly after the drowsiness level reaches one hundred. The percentage will decrease slightly by ten percent at this stage. Also, the authors give a chance to the driver by setting the drowsiness level to fifty percent provided the driver wakes immediately after the alarm has been activated.

As the authors assume that one hundred percent drowsiness level indicates a high chance for the driver being drowsy, the authors chose to alert the driver by sound. The authors chose the 'Siren' sounds as it is one of the most annoying sounds and can easily grab attention.

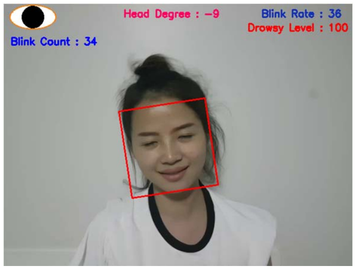

Figure 10. The Drowsy Level Reaches One Hundred

G. Hardware

The Raspberry Pi 3 Model B and Raspberry Pi camera are used in this paper because they are small in size. Actually, the authors plan to set them in a car, but the paper requires constant light conditions for better performance. Also, a higher frame rate that supports 1080p30, 720p60 and 640×480p60/90 is also necessary [16]. The resolution of the camera is five megapixels. In addition, it is fast enough to capture in real time. Therefore, the authors only installed them by using a piece of carton. Another advantage of Raspberry Pi 3 deals with the high performance of its CPU compared with other Raspberry Pi. This paper uses real-time processing because a driver can fall asleep while driving. Thus, the authors needed a faster CPU performance in order to check whether or not the driver is drowsy. Moreover, the authors used Raspberry Pi 3 Model B because it is affordable. It is around two thousand five hundred baht for both the Raspberry Pi 3 and its camera.

H. Software

The researchers in this paper decided to use Raspberry Pi 3 Model B and Raspberry Pi Camera in because of the high performance of its CPU and higher frame rate. Raspberry Pi 3 Model B supports C++ and OpenCV library. Also, the paper is done by using Microsoft Visual Studio 2015. The authors apply OpenCV Version 3.1.0 for various features of computer vision. The Haar Cascade Classifier, warpAffine and template matching are supported in OpenCV library.

III. Experiment Results

A. Different Skin Colour Implementation

The authors implemented innovative algorithms with ten people, comprised of five men and five women. They were tested for eye blink rate as well as head tilting. In order to make sure that this paper was capable of detecting the face and eyes of all people, the authors checked for various skin tones and different light conditions. The solution to different skin colour is template matching. The authors conducted the paper with two skin tones, including white skin and tan skin, which worked successfully with the algorithms.

B. Different Light Condition Implementation

The hardest part in this paper dealt with handling different light conditions. One of the reasons was that this paper used the Haar Cascade Classifier, which is not able to detect the face of people in dark areas. It is trained for bright conditions. If people stay in the dark room, the light conditions are not stable. This possibly causes noises in the frame. The authors also tested the algorithms in different areas where the light was not constant. The methods used to handle different light conditions were template matching and region of interest.

C. Wearing Glasses Implementation

The efficiency of the Haar Cascade Classifier is decreased when it is used with a person wearing eyeglasses. One of the reasons is light reflection from the eyeglasses, according to Figure 11. However, the authors tested a person who wears glasses, which was able to detect the driver in this paper in every frame.

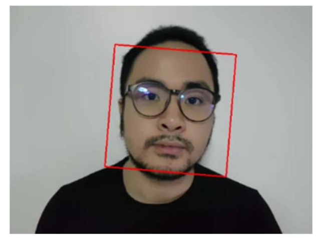

Figure 11. Test of Person Wearing Glasses

D. Drowsy Stage Implementation

The combination of face detection and eye detection brings forward the drowsiness check process. There are three stages of the drowsiness level. If the drowsiness level is less than forty percent, the screen will display text in green. A driver tends to fall asleep while driving. As such, the next stage is drowsiness level between forty and eighty percent, in which text will be shown in yellow. The driver is most likely to fall asleep during this stage.

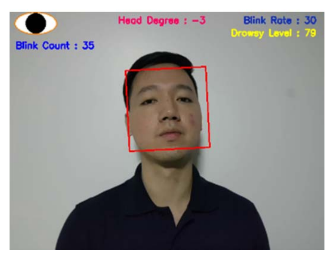

Figure 12. Yellow Stage of Drowsiness Level

The final stage of the drowsiness level will be shown in red. This means that the drowsiness level is more than eighty percent. This is considered very dangerous. In this case, the driver should stop driving because of the potential for falling asleep. The driver might frequently blink his eyes and tilt his head. In addition, the authors tested this paper with ten test cases. The paper was able to calculate eye blink and head tilting accurately. Therefore, they are checked whether or not they are drowsy.

E. Hardware Implementation

At the beginning of this paper, the authors experimented in a bright room where the light was constant. Ten test cases were recorded by mobile phone. An example of such recorded video on mobile phone is shown in Figure 12. Afterwards, the authors applied the code in Raspberry Pi 3 Model B and Raspberry Pi Camera.

IV. Results and Evaluation

The authors found that the tested accuracy rate for face detection and eye detection from the group of ten volunteers was effective, as shown in Table II. There are some definitions of true positive, false positive, and false negative that need to be mentioned. Firstly, true positive means the number of times that the algorithm correctly detects eye blinking of the driver. Secondly, false positive can be described as the number of times that the algorithm counts the value of eye blink when the driver does not blink. Lastly, the meaning of false negative deals with the number of times the algorithm does not count the value of eye blink when the driver actually does blink.

According to Table II, the number of false positives and false negatives was very low when the authors applied the methods to check eye blink, which included region of interest, template matching and geometric rotation. The number of true positives is very close to the total blink. In addition, the authors tested with ten volunteers, so the formula for recall and precision is respective [17]. The value for recall is 99.59 percent, whereas the value for precision is 97.86 percent.

TABLE II. Accuracy Rate of Face and Eye Detection

Gender	Accuracy Rate of Face and Eye Detection			
	Total Blinks	*True Positive (%)*	*False Positive (%)*	*False Negative (%)*
Male 1	95	100.00	0.00	0.00
Male 2	103	99.03	0.00	0.97
Male 3	100	99.00	1.00	1.00
Male 4	83	98.80	1.20	1.20
Male 5	88	100.00	11.36	0.00
Female 1	99	98.99	0.00	1.01
Female 2	100	100.00	3.00	0.00
Female 3	100	100.00	1.00	0.00
Female 4	100	100.00	0.00	0.00
Female 5	98	100.00	5.10	0.00
Total	**966**	**99.59**	**2.17**	**0.41**

The authors calculated the total number of frames between the Haar Cascade Classifier and the Haar Cascade Classifier with the algorithms. As observed in Table III, the proposed work caused an increase in accuracy for face detection by sixteen percent.

Another three test cases that the authors checked was for full close. If the driver closed the eyes for more than two seconds, the authors considered it as full close, as seen in Table IV. The percentage of full close that this paper could check was one hundred percent. Moreover, the Haar Cascade Classifier can accurately detect different skin colour on the face.

TABLE III. Comparison between the Haar Cascade Classifier and this Paper

Gender	Comparison between Haar Cascade and this Paper			
	Total Frame	*Haar Cascade*	*Proposed Work*	*Difference*
Male 1	5,236	4,660	5,236	576
Male 2	3,139	2,509	3,139	630
Male 3	4,063	3,270	4,015	745
Male 4	2,330	1,992	2,330	338
Male 5	2,647	2,216	2,644	428
Female 1	3,289	1,862	3,289	1427
Female 2	3,317	2,733	3,317	584
Female 3	3,512	3,149	3,512	363
Female 4	2,853	2,567	2,853	286
Female 5	3,570	3,255	3,570	315
Total	**33,956**	**28,213**	**33,905**	**5,692**

TABLE IV. Percentage of Full Close

Gender	Percentage of Full Close			
	Total Full Close	*True Positive*	*False Positive*	*False Negative*
Male 1	10	10	0	0
Male 2	10	10	0	0
Male 3	10	10	0	0
Total	**30**	**30**	**0**	**0**

As the results show in Table II, the current work could detect a face with 99.85 percent of the total frame. Also, the paper tested ten people under constant light conditions and two people under inconsistent light conditions. It works with face and eye detection under twelve such conditions. The Haar Cascade Classifier could detect both of them in every frame. Additional implementation was tested with a person wearing eyeglasses, which was also able to detect the face and eyes.

V. Conclusion

Driver Drowsiness Detection was built to help a driver stay awake while driving in order to reduce car accidents caused by drowsiness. This paper experimented in a bright room with constant light. In addition, there were several limitations including light conditions and the darkness of the skin. This paper was concerned with drowsy drivers and their potential to cause car accidents. Therefore, the authors applied image processing and C++ language skills in order to build an embedded device that could alert drivers when feeling sufficiently drowsy. The embedded device can calculate a drowsiness level from the driver using a combination of Raspberry Pi 3 Model B and Raspberry Pi Camera. Raspberry Pi 3 Model B is a processor to calculate whether or not a driver is drowsy. At the same time, it retrieves images from the camera, which is fast enough to detect a driver’s features in real time.

A solution to calculating the drowsiness level of a driver is to detect the face, eye blinking, and head level.

Face detection is done using the Haar Cascade Classifier, which is trained for thousands of people with different light conditions. The position of the head must be straight, after which the eyes can be detected for being opened or closed by using template matching. However, the Haar Cascade Classifier is only trained for upright faces, so it is possible for a driver to tilt his or her head when feeling drowsy. One way to solve this problem is to use geometric rotation. If a driver tilts the head, the frame can still be rotated into an upright face by calculating the angle. Further, eyes closeness and the rate of blinking can be calculated easier. In addition, the level of drowsiness can be checked after accurate detection of the eyes and face in every frame. This is accomplished through calculation of the frequency of eye blinking and head tilting. If the driver blinks too frequently, drowsiness is indicated. When the drowsiness level reaches one hundred, the driver will be alerted by a loud audible warning, which can wake the driver up before falling asleep while driving.

Future development for this research involves installation of the device in a car. The developers will be able to test for many variables such as light conditions and the drowsiness level of the driver. After the developers test the device in a car, it is anticipated that any design flaws can be identified and more data can be gathered concerning the practical operation of the device in real situations.

ACKNOWLEDGMENT

This project was supported by the Mahidol University International College and the Department of Computer Engineering, Faculty of Engineering, Mahidol University.

REFERENCES

[1] Association for Safe International Road Travel (ASIRT), Road Crash Statistics. http://asirt.org/initiatives/informing-road-users/road-safety-facts/road-crash-statistics, 2016.

[2] Bangkok Post. (2015, October 20), Thailand's roads second-deadliest in world, UN Agency finds. http://www.bangkokpost.com/archive/thailand-roads-second-deadliest-in-world-un-agency-finds/736748, 2016.

[3] Jan, T., Karnahl, Seifert, K., Hilgenstock, J., & Zobel, R. Don't sleep and drive – VW's fatigue detection technology. http://www-nrd.nhtsa.dot.gov/pdf/esv/esv19/05-0037-O.pdf,2016 .

[4] Drowsy Driving, Facts and Stats: Drowsy Driving – Stay Alert, Arrive Alive. http://drowsydriving.org/about/facts-and-stats/, 2016.

[5] J. Qiang and X. Yang, "Real-time eye, gaze, and face pose tracking for monitoring driver vigilance," *International Journal of Real-Time Imaging*, vol. 8, 2002, pp. 357–377, doi:10.1006/rtim.2002.0279.

[6] Hong, Tianyi, Huabiao Qin, and Qianshu Sun. "An improved real time eye state identification system in driver drowsiness detection," *in Proc. IEEE Int. Conf. on Control and Automation*, IEEE, 2007, pp. 1449-1453.

[7] S. Hu and G. Zheng, "Driver drowsiness detection with eyelid related parameters by Support Vector Machine," *International Journal of Expert Systems with Applications*, vol. 36, 2009, pp. 7651–7658, doi: http://dx.doi.org/10.1016/j.eswa.2008.09.030.

[8] Warning Message via Display Monitor in VW's Car, VW Golf Turan some improvement. Driver fatigue detection system as standard | Responsejp (Automobile new model / new models). http://en.responsejp.com/article/img/2012/07/26/178559/367973.html, 2016.

[9] Emami, S. (2010, June 2), Face Detection and Face Recognition with Real- time Training from a Camera. http://www.shervinemami.info/faceRecognition.html, 2016.

[10] Face Detection using Haar Cascades. OpenCV: Face Detection using Haar Cascades. http://docs.opencv.org/master/d7/d8b/tutorial_py_face_detection.html#gsc.tab=0, 2016.

[11] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *in Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2001, pp. 511-518.

[12] P. Viola and M. Jones, "Robust real-time face detection," *International Journal of Computer Vision (IJCV)*, vol. 57, 2004, pp. 137-154.

[13] Template Matching—OpenCV 2.4.13.0 documentation. http://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/template_matching/template_matching.html, 2016.

[14] Utility and System Functions and Macros—OpenCV 2.4.13.0 documentation. http://docs.opencv.org/2.4/modules/core/doc/utility_and_system_functions_and_macros.html#gettickcount, 2016.

[15] Stormberg, J. (2012, December 24), Why Do We Blink So Frequently? | Science | Smithsonian. Retrieved from http://www.smithsonianmag.com/science-nature/why-do-we-blink-so-frequently-172334883/?no-ist, 2016.

[16] Raspberry Pi Camera Board, Raspberry Pi Camera Board ID: 1367 - $19.95: Adafruit Industries, Unique & fun DIY electronics and kits. https://www.adafruit.com/product/1367, 2016.

[17] Sensitivity and specificity, Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Sensitivity_and_specificity, 2016.