

The System for Driver Fatigue Monitoring Using Decision Tree via Wireless Sensor Network for Intelligent Transport System

<https://doi.org/10.3991/ijoe.v14i10.7507>

Worawut Yimyam, Mahasak Ketcham^(✉)
King Mongkut's University of Technology North Bangkok, Bangkok, Thailand
mahasak.k@it.kmutnb.ac.th

Abstract— Researchers propose new algorithm to monitor fatigue of driver by tracking the movements of the eyes. This can classify the opening and closing eyes by using Haar-Like Features and Region of Interest techniques. The fatigue can be classified by decision tree classification. The system works in real time and sends message to the mobile phone to alert the driver via Line Application. The results showed that the program can detect face and classify opening and closing eyes accurately at 99.93 percent in which it yielded higher accuracy than other algorithms.

Keywords— Fatigue, Region of Interest, Decision tree

1 Introduction

Improvement of safety system for people and reduction of accidents are important goals for Intelligent Transportation System (ITS). Key factor causing accidents on high-ways is fatigue and tiredness while driving. Fatigue reduces ability to control a car. The study showed that a driver can drive a car 2-3 hours straight. After that, a driver will be tired after lunch, in the afternoon, after midnight, after drinking alcohol and taking medicine that causes drowsiness reducing driving performance [1][2]. From psychologists and physiologists' points of view, fatigue, drowsiness and distraction are different. However, from the view of Intelligent Transportation System (ITS), fatigue, drowsiness and distraction have the same definitions. In terms of science, no criteria can measure fatigue accurately [3][5]. There are 3 types of fatigue; 1) Sensory Fatigue; 2) Muscle Fatigue and; 3) Cognitive Fatigue. Sensory Fatigue and Muscle Fatigue are measurable but Cognitive Fatigue is not. Certain meaning is not identified. However, a relation between fatigue, drowsiness, body temperature, eye movement, respiratory rate, heart rate and brain function is found [2][5][6][7]. The most effective way to measure fatigue and drowsiness is to measure brain function. This method requires signal from brain and electron wires must be connected to the head of an examined person which disturbs and intrudes upon a person. After brain function examination, fatigue will appear in eyes. According to research studies, visual stimulation and response is key measure to examine consciousness called Psychomotor Vigilance Task (PVT). Therefore, this

research shows that PVT is related to movement of eyelids at a certain period of time and percentage of eyelid closing at PERCLOS [4].

Based on previous studies, technologies have been applied to manage fatigue and drowsiness in the field of Intelligent Transport system (ITS) [2] to examine a driver's tiredness which causes drowsiness while driving, leading to accidents on highways. Scientific technologies are used to measure fatigue but it cannot give accurate and precise results [6][7]. At present, there are plenty of research studies and development methods to measure fatigue, eye functions and eye blinking. According to related research, measuring and tracking fatigue in soldiers still has problem with accuracy in monitoring and timely tracking of fatigue which possibly causes accidents. Moreover, fatigue cannot be predictable in advance so it cannot reduce any losses. Therefore, the researcher is interested in developing driver fatigue monitoring algorithm using position detection and eye movement techniques to measure fatigue. The researcher also utilizes Haar – Like and Region of Interest (ROI) to detect position and movement of eyes to analyze fatigue through Decision Tree algorithm and to monitor fatigue.

2 Related Research

The study of Hoang Le, Thanh Dang and Feng Liu [8] proposes eye blinking detection by intelligent glasses that classified features of eye blinking, eye opening and eye closing, using Gradient Boosting (GB) for data training. Then, a comparison of images captured by the intelligent glasses was carried out. However, it found that images were sometimes dark so the system could not classify the data. The researchers added eye position technique by using Haar Cascade from OpenCV to increase data processing efficiency. The results were satisfied.

YenWei Chen and Kenji Kubo [9] studied and developed face detection and quick eye movement through a webcam, using Gabor Filter technique. This technique combined a filter with color data to detect face area in geometric shape on human facial structure. Then, Gabor Filter detected all facial movement positions. The system performed detection continuously and showed results of face areas detected on a monitor. The results showed that the system could detect face accurately.

Sung-Uk Jung and Jang-Hee Yoo [10] proposes a way to increase eye detection efficiency by eliminating disturbance by using SQI. In addition, a conversion of image into 3D-format could identify position of eyes accurately. Using AdaBoost to analyze data also identified eye position effectively.

The study of Ziad O [11] was developed to assist patients or disable persons. The study develop a communication system through eye control, using Matlab version 6.5 to calculate positions expected to be eyes. Images were compared to those in the program's database to examine image detection, using special algorithm of Matlab. After eye movement was detected, an image was shown on a monitor of the program as cartoon or different positions. The system compared eye movement with position on a monitor where eyes stared on. Then, a sound or a message showing meaning of image would appear. The accuracy of results was at 96.11 % and the accuracy of repeated performance after the first time was at 94.44%. However, there are many factors

affecting the system performance in actual system development. For instance, a camera shot angle should cover a focus range or other movements not related to eye movements such as head movement of an examined person should be reduced etc.

Vidas Raudonis, Rimvydas Simutis and Gintautas Narvydas [12] developed a measurement tool. They connected a pair of glasses to a web-camera to have more convenience and to reduce distance that enables more effective eye movement detection. Artificial Neural Network and MATLAB were applied to classify features of eyes in each group to detect eye position. The results showed that the tool worked effectively. It could classify elements for eye tracking development. However, there were some limitations. The system had slow response when a user changed eye direction too quickly. The outside factors such as light affected data analysis.

Gerald Fritz and Lucas Paletta [13] investigated the use of Eye Tracking technology to analyze marketing research about creating advertising media or videos that would keep people interested in the advertisements. They studied behavior in looking and paying attention to different billboards at public transportation areas. The researchers used an eye tracking program to identify the most frequently looked points. The data was collected at public transportation areas in Graze, Austria. The study investigated behavior and interest in the most frequently looked points when using public transportation. The most frequently looked points were at televisions, moving points or advertisement images or at 77%.

N.M.Mohd Hassan and W.Mansor [14] studied about eye movement detection to control TV-switching on and off from the distance through an analysis of EOG (Electro Oculography). This technique applies electrical potential measurement of retina. A signal from the measurement is called Electro Oculogram. The researchers carried out the experiment with 9 normal visual ability volunteers. Five electrodes around an eye were connected (4 points on the right and 1 point on the left). The other ends were connected to a computer to record signals and for processing. The results showed that the accuracy was at 90.44%. However, this method required hardware with microcontroller system which could be connected to a television.

The study of Mehrube [15] investigated algorithm development for increasing eye tracking efficiency to control in-house safety systems for patients. The study developed a program on smart camera through OpenCV which was Open source and Visual C. The results showed that the program could work effectively. It could track eye movements accurately. Nevertheless, when images moved or had obstructions, for example, there were glasses or some parts of a face were missing or a distance between eyes and a camera was not proper, the processing would reduce.

2.1 Decision Tree Generation

After generating a decision tree, it can be used to classify categories of the data recorded (which consists of different attributes but data categories cannot identified) by comparing attributes in the particular record. A comparison begins from root nodes to left nodes. When reaching leaf nodes, data categories being considered are identified.

During late 1970s, the researcher about machine learning named J. Ross Quinlan invented decision tree generation algorithm known as ID3 (Iterative Dichotomiser).

After that, he developed ID3 to be C4.5 which became foundation algorithm used for comparing algorithm performance efficiency of Supervised Learning.

ID3 and C4.5 apply greedy approach to generate a tree under the concept of “top-down recursive divide-and-conquer”, considering training data (training data, record set of data consisting of set of attributes and attributes indicating data categories of record set of data) by partitioning data into small parts while generating a tree .

3 Proposed Method

Elements of fatigue detection tool.

3.1 Driver Fatigue Monitoring Hardware

The structure of the driver fatigue monitoring system consists of 2 main parts including a camera module and a Raspberry PI 3 model B board. This board is powered by 5-volt power supply as shown in this flowchart.

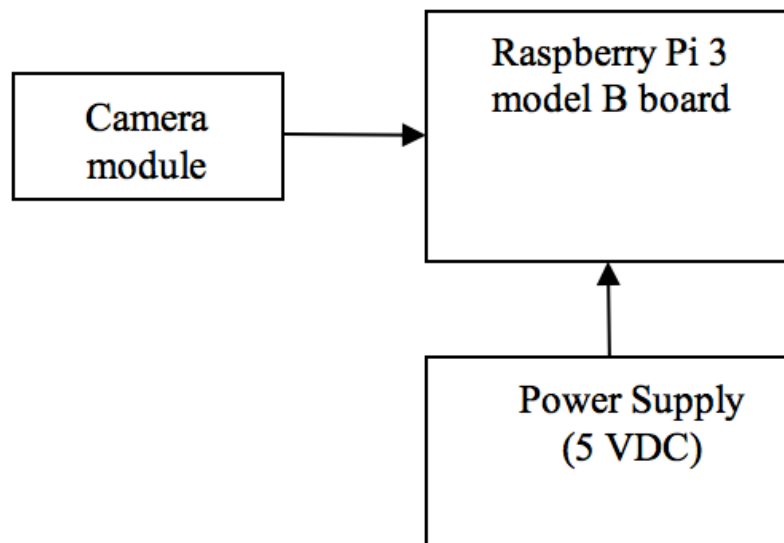


Fig. 1. Diagram of driver fatigue monitoring system hardware

Details and functions of the 2 main parts are as follows.

Camera module. A camera module receives a real-time motion picture while driving and transmits the frame to a Raspberry PI 3 model B. Picture resolution depends on camera resolution settings.

Raspberry PI 3 model B board. A Raspberry PI 3 model B board controls the camera module and its processing. The Raspberry PI 3 model B board is designed to receive 5-volt power supply directly.



Fig. 2. Raspberry Pi 3 model B board and camera module

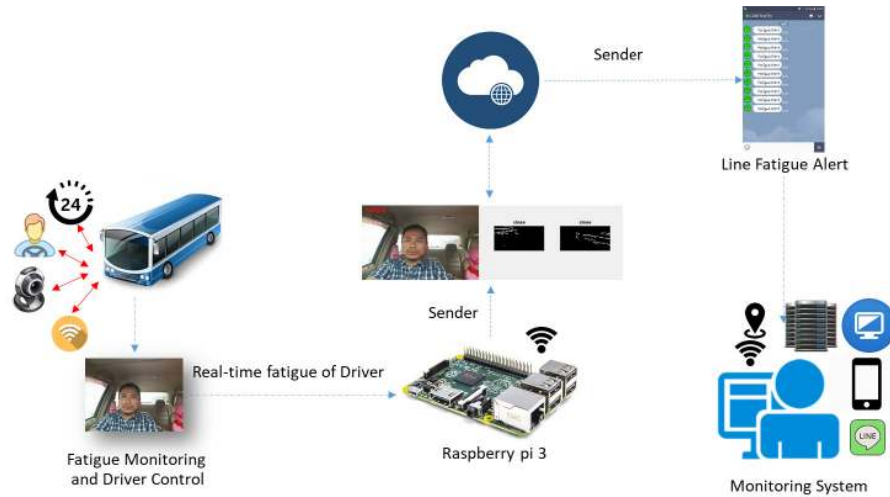


Fig. 3. Flowchart of driver fatigue monitoring system

Fig.3. shows the driver fatigue monitoring system for public drivers in charge of transferring passengers on public buses. There is a camera on a bus to receive real-time motion pictures of a driver. Then, the motion pictures are transmitted to the Raspberry Pi board that controls the camera and the processing of the driver fatigue monitoring system when a public driver is tired and drowsy. The system monitors and tracks eye functioning. When a driver’s two eyes including a left eye and a right eye are closed, the system will send an alert to inform a monitor that a driver is drowsy. The Monitoring System will monitor, track, make a plan and change a public driver to reduce accidents promptly.

3.2 Flow Chart: Eye Tracking Process & Fatigue

Eye movement detection process is as follows.

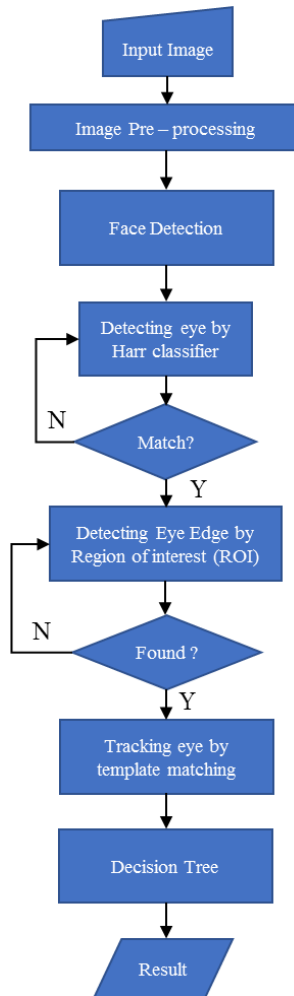


Fig. 4. Eye Tracking Process and Fatigue.

Input Image from Webcam. When a user put glasses on, a camera will detect a user's eye image by using HighGUI command.

Gray-scale (Image Pre-processing). An RGB image is converted to a gray-scale image as follows.



Fig. 5. Eye Tracking Process and Fatigue.

Eye position is detected by Haar Classifier.



Fig. 6. Eye position detection ClassifierCascade

Detecting Eye Edge by ROI. Region of interest (ROI) is applied to cover region of interest. Then, a circle is create on a black eyeball identified as the middle of the black eyeball (x,y). A radius (r) is identified by referring to an equation for circular area calculation as follows.

$$\text{Circular radius}(r) = \sqrt{(x - \text{Center}_x)^2 + (y - \text{Center}_y)^2} \tag{1}$$



Fig. 7. Sample of Identifying Region of interest (ROI).

Tracking Eye by Template Matching. It is a process to track eye movements to monitor fatigue through eye opening and eye closing. It can be calculated to identify a starting coordinate for movement as follows.

$$(x_c, y_c) = \left(\frac{x_r + x_l}{2}, \frac{y_r + y_l}{2} \right) \tag{2}$$

When

(x_c, y_c) = A middle coordinate between two eyes

(x_r, y_r) = A coordinate of right eye

(x_l, y_l) = A coordinate of left eye

When a starting coordinate is identified, a distance created by changing positions of the starting coordinate by a user will affect cursor movements. The distance can be calculated by using the equation below.

$$\text{distance} = \sqrt{(x_c - x'_c)^2 + (y_c - y'_c)^2} \tag{3}$$

When

$$x_c, y_c = A \text{ middle coordiante of black eyeball}$$

New coordinate where a mouse moves onto can be calculated by using the equation below.

$$\theta = \arctan\left(\frac{y_c - y'_c}{x_c - x'_c}\right) \quad (4)$$

$$x_{c_update} = x_c + \frac{distance}{3} \cos\theta \quad (5)$$

$$y_{c_update} = y_c + \frac{distance}{3} \sin\theta \quad (6)$$

$$(x_{c_update}, y_{c_update}) = A \text{ coordinate of a mouse} \quad (7)$$

This research combines MatchTemplate of OpenCV with MotionEvent on Android.

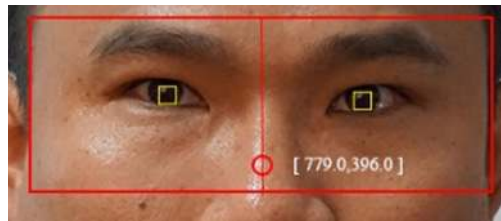


Fig. 8. Sample of Identifying Starting Coordinate for Movement Reference.

Face Detection. The system monitors that whether the image contains face position by using library of CascadeClassifier of OpenCV. It compares histogram from left eye and right eye positions. If a face is detected, a rectangle will be formed around face area. On the other hand, a face is not detected within duration specified.

Get New Frame. Eye Detection is shown in Fig 9. Match template function of OpenCV is operated with Motion Event on android mobile.

Resizing eyes. An image from the previous step is resized to the standard size because an image captured by the camera come in different sizes. It is required to resize sizes of eyes to be at the same standard for precise extraction of key features of the eye in the next process. In this study, the image standard size is 24*24 pixels.

Tree Classification. Classifying data by using decision tree is a process of creating a tree for making decision based on data with data categories. A decision tree consists of different nodes (which are non-leaf nodes) used to display a certain condition or an attribute of data. Each branch of node refers to possible value from the test carried out with the particular attribute. The tree also consists of left node with data categories are stored. A sample of decision tree is shown in Fig. 10. Sample of a decision tree for open eye and close eye classification.

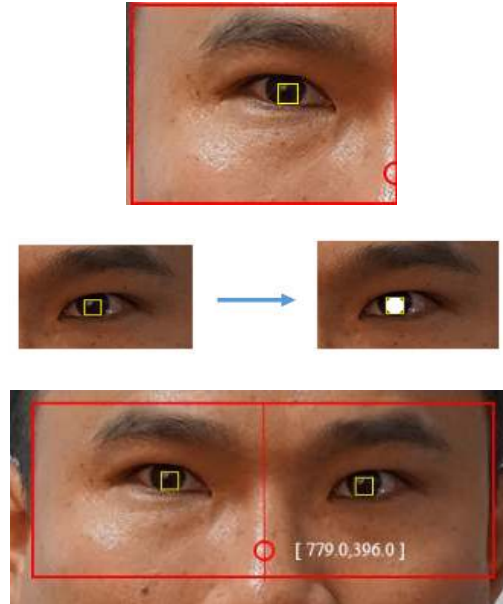


Fig. 9. Eye Detection.

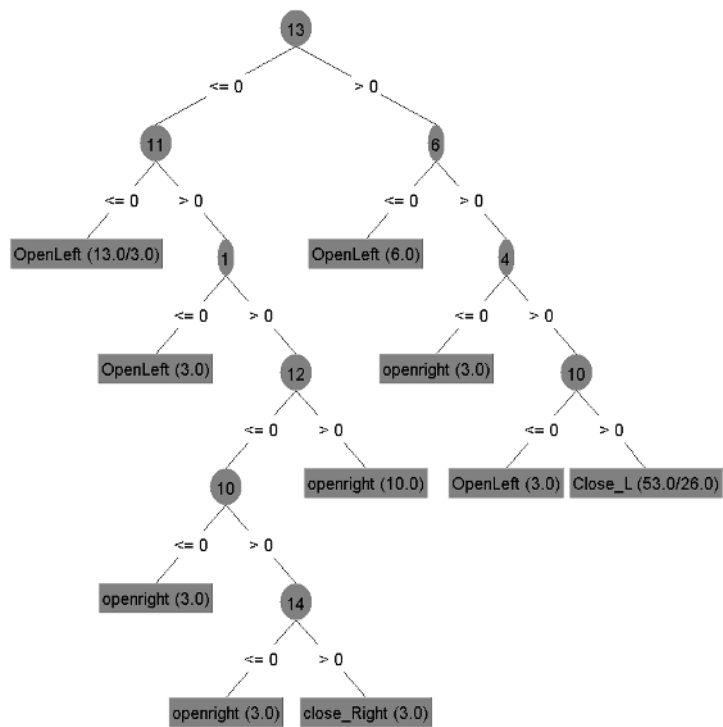


Fig. 10. Sample of a decision tree for open eye and close eye classification

3.3 Data transmission

When the system begins to work, the camera will work with the Raspberry Pi board to monitor a public driver's fatigue by algorithm of fatigue monitoring. When the system detects a public driver's fatigue, the Raspberry Pi board will send an alert to the Monitoring System. The researcher has designed the system that can send data through Line Messages, SMS Text Messages and GPS, using Application Programming Interface to work with the Raspberry Pi and 4G Shield GPS, which connects to 4G as shown in Fig.11.

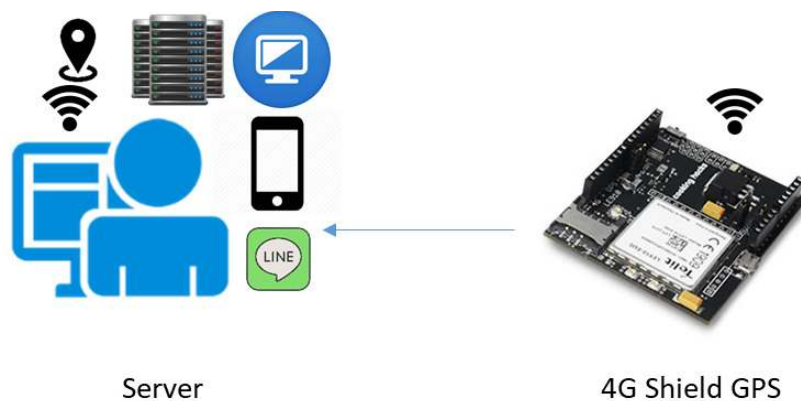


Fig. 11.Data transmission of fatigue monitoring system

Fig.11. shows that the system will connect to 4G mobile network that works with the Raspberry Pi board to monitor a driver's fatigue and send data via 4G to a server.

When the system sends the data of a driver's fatigue to the server, the server will keep the data in a database. The data is pictures of a driver with fatigue while driving. A system monitor can then monitor the data via data management program and monitor program. A monitor program will display a bus location and an alert of fatigue.

The data from the system detects a driver's fatigue and is transmitted to the server. The data is stored and processed when a driver's fatigue is detected. Then, the system sends an alert to the Monitoring System where a monitor will change and inform a driver to reduce accidents.

3.4 Network processor

A network processor of fatigue monitoring system has a module to create a network system via 4G and it is installed in each public bus to monitor a driver's fatigue. The system processes the data to monitor a driver's fatigue and sends the data to a server as shown in Fig 12.

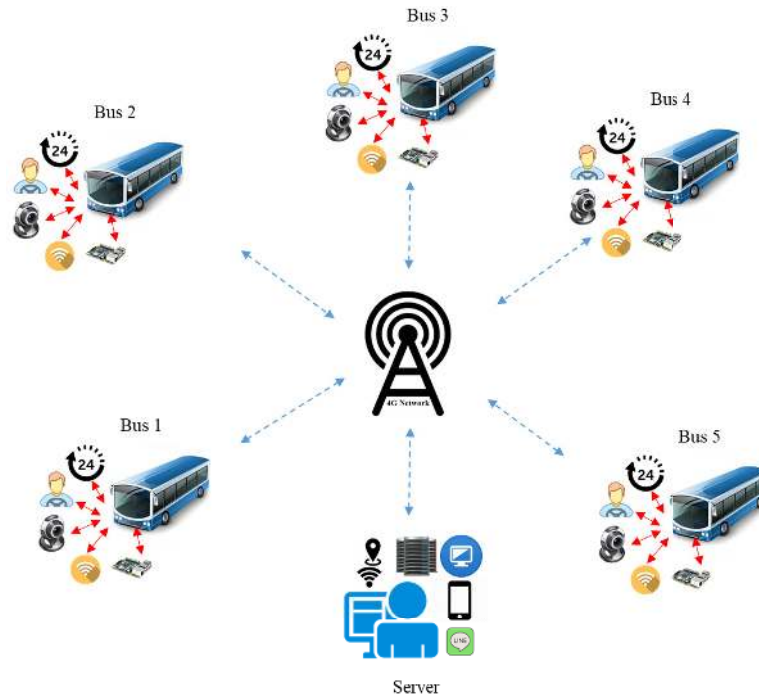


Fig. 12. An overview of a driver fatigue monitoring system for public drivers

4 Experimental Results

4.1 Eye Detection Testing

Eye detection testing processes the image receiving and location tracking procedures while starting the program

4.2 Eye Movement Testing

Eye movement testing verifies the directions of the eye movement including,

- The system tracks at the left side when the eyes turn left.
- The system tracks at the right side when the eyes turn right.
- The system tracks at the topside when the eyes move to above.
- The system tracks at the bottom side when the eyes move to below.

4.3 Face Detection Testin

Face detection testing performs the system when users move their head. Its performance testing for analyzing the accuracy is shown in equation (8)

$$A = \frac{N_c}{N} \times 100 \tag{8}$$

A is percentage of accuracy.
 N_c is number of accurate processing.
 N is number of all the experiments.

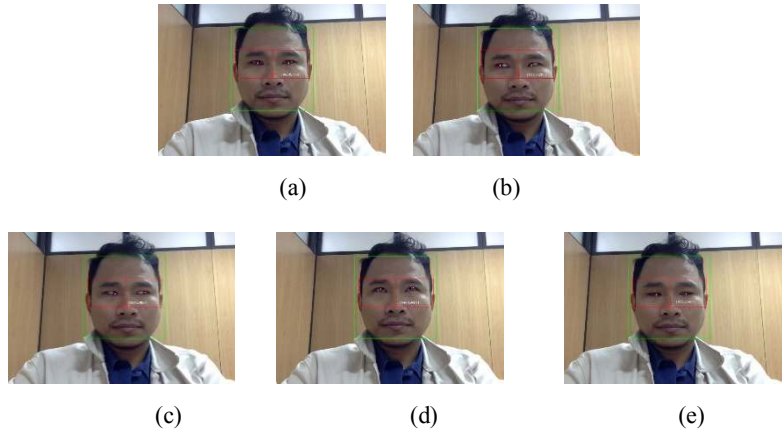


Fig. 13. (a) The experiment of the face detection and eye tracking at the straight view. (b) The experiment of the face detection and eye tracking at the left view. (c) The experiment of the face detection and eye tracking at the right view. (d) The experiment of the face detection and eye tracking at the top view. (e) The experiment of the face detection and eye tracking at the bottom view

The experiment is divided into two types; performance testing for the eye detection and eye tracking. The results are shown as below.

Table 1. Experimental results of the performance of the eye detection

	The position of the eye detection					Mean
	<i>Straight</i>	<i>Top</i>	<i>Bottom</i>	<i>Left</i>	<i>Right</i>	
Accuracy	0.96	0.87	0.89	0.96	0.95	0.93

Table 2. Experimental results of the performance of the eye tracking

	The position of the eye tracking					Mean
	<i>Top</i>	<i>Bottom</i>	<i>Left</i>	<i>Right</i>	<i>Straight</i>	
Accuracy	0.73	0.76	0.69	0.75	0.98	0.78

From the experiments that have tested for 100 times, we found the accuracy of the eye detection and eye tracking performances are described as follow;

The accuracies from the eye detection capability when the eyes view straight on, look at the top, look at the bottom, turn to the left, and turn to the right are 96, 87, 89, 96, 95, and 93 percent respectively. The accuracies from the eye tracking capability

when the eyes look at the top, look at the bottom, turn to the left, turn to the right, and view straight on are 73,76,69, 98, and 78 percent respectively.



Fig. 14.Results of the Eye Detection.

The researcher uses 2,423 images captured from human in the study. The images included 4,846 eye-closing and opening images divided into 1,192 left-eye closing images and 1,192 right-eye closing image and the eye-opening images were 2,462 divided into 1,231 left-eye opening images and 1,231 right-eye opening images.

Table 3.

Model	Accuracy	Precision	Recall	F-Measure
Proposing methods	99.93	0.99	0.99	0.99
Logistic	98.7	0.987	0.987	0.987
Fuzzy	96.80	0.971	0.968	0.968
ANN	97.28 %	0.973	0.973	0.973
Navebaye	73 %	0.681	0.731	0.670
AdaBoots	49.91 %	0.499	0.499	0.427
ZeroR	25.38	0.129	0.254	0.137
Vote	25.38	0.129	0.254	0.137

According to Table 3, the most accurate method was Decision Tree method. The accuracy was at 98.68%, the Precision was at 99%, the Recall was at 99% and the F-Measure was at 99%. The lowest accurate method was ZeroR and Vote. The accuracy was at 25.38%, the Precision was at 12.9%, the Recall was at 25.4% and the F-Measure was 13.7%. Therefore, the Decision Tree method was the most accurate method for analyzing fatigue by classifying eye-closing and eye-opening performance. The classification covered the left-eye closing image, the right-eye closing image, the left-eye opening image and the right-eye opening image shown as follows.

Figure 15-16 show eye-closing and eye-opening performance which indicate fatigue. It considers the closing and opening performance of the eyes. When the system detects eye-closing, it alerts that there is fatigue. On the other hand, when the system detects both eye-opening, it informs that there is no fatigue. The system will send a notification to a mobile phone through Line Message. It is found from the experiments that the system can detect eye movement from all face positions. Figure 17 shows a notification through Line Message.



Fig. 15. Experimental Results of Eye-Closing Classification.

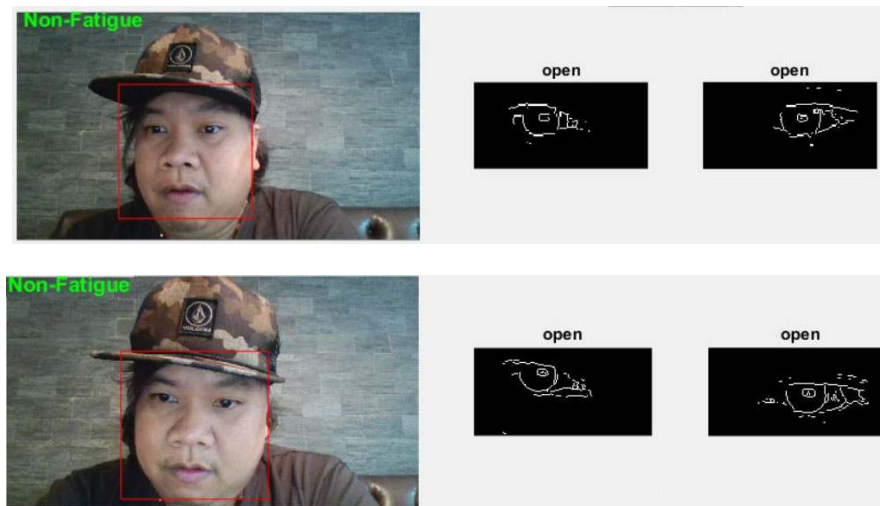


Fig. 16. Experimental Results of Eye-Opening Classification.



Fig. 17. Show a notification through Line Message.

4.4 Installation and Testing

The researcher installed the device and modeled an actual use, using the researcher's car. Installation details are as follows. 1) Camera Logitech is positioned in the front part of the car and a driver seat. The distance from a driver is 70 centimeters to detect a driver's face. 2). Raspberry PI 3 is positioned on a car's console and is connected to power supply and USB Car to test (Figure 18). To test the device, the researcher closed the eyes and drove the car. The system detected and sent a notification to the center and to the driver's smartphone and a control center.

4.5 Test of Notification Sent to Smartphone

When the system detects a driver's fatigue, the system will send a notification to a smartphone via Line Application. In this research, the researcher drove a car to test fatigue through closing the eyes for 30 times. Each time took 2 seconds or more. The system sent notifications as shown in Table 4.



Fig. 18. Fatigue Monitoring System Installation

Table 4. Test of Notification Sent to Smartphone

Fatigue	Times	Number of Accurate Sent Notifications	Number of Inaccurate Sent Notifications	Accuracy (%)
Fatigue	30	30	0	100
Total	30	30	0	100

It was found that the system could send accurate notifications notifying the driver's fatigue through eye closing. The monitoring result was 100 percent accurate. Sample of notification sent is shown in Figure 19.

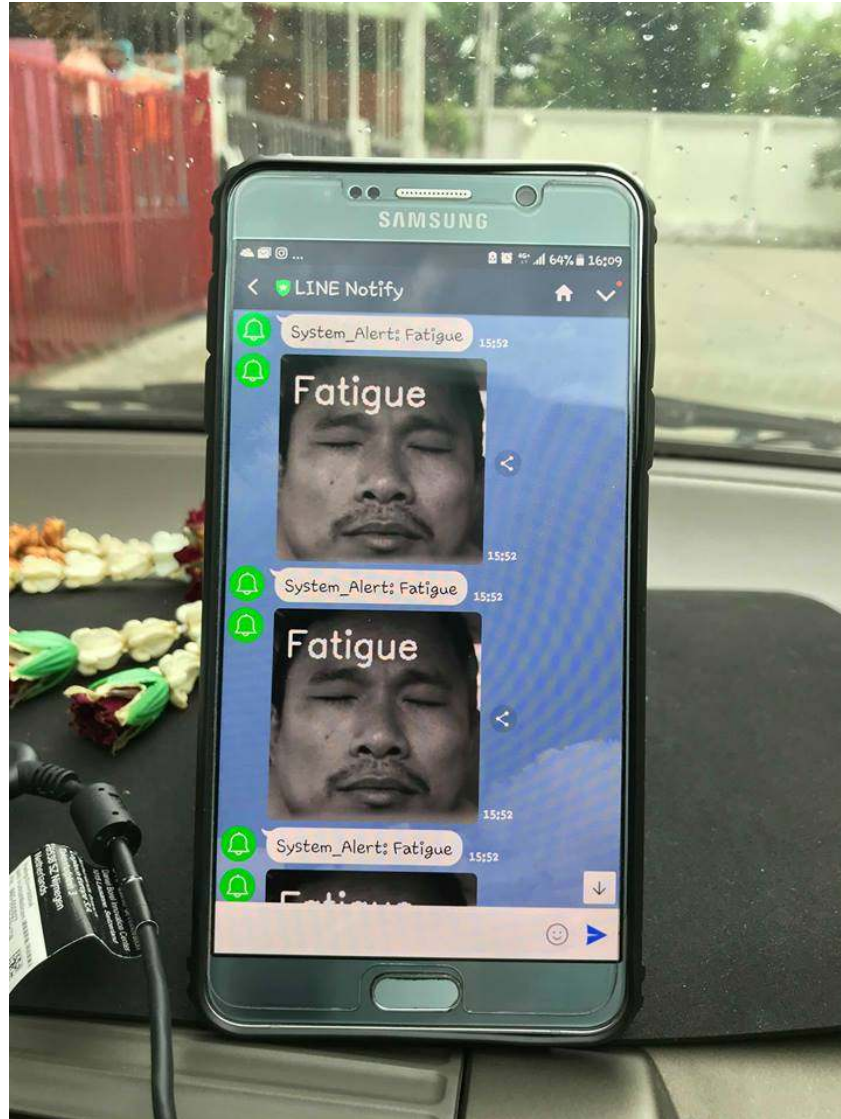


Fig. 19. Sample of Notification Sent to Smartphone

5 Conclusion

The researchers proposes new algorithm for fatigue tracking system by using position detection and eye movement to detect fatigue. Haar-Like classifier and Region of Interest were employed to detect position and eye movement and to analyze fatigue through Decision Tree. The experimental results showed that the performance of the algorithm could detect face, eye positions and fatigue accurately. The accuracy was at

99.93 percent. This algorithm can be applied in Real Time usage because the system can send an alert through Line Message. The system detects fatigue and can be used with public drivers. The system has been designed to have a camera to receive pictures to be processed on the Raspberry Pi boards and to have the system connected to 4G network module and installed in each public bus in order to detect a public driver's fatigue. The system tracks a driver's eye functioning. When a driver's left eye and right eye are closed, it demonstrates that a driver is tired. Then, the system will transmit the data to the server via 4G. The data to be transmitted includes pictures of a left eye and a right eye while closing, a public bus location and an alert of fatigue etc. The system also sends an alert via Line message to involving persons including a system monitor, a police officer, a driver etc. in order to use in planning and changing a public driver, which can reduce accidents.

6 References

- [1] Haworth, N. L., Triggs, T. J., and Grey, E. M. (1988). Driver fatigue: Concepts, measurement and crash countermeasures (No. CR 72).
- [2] Lin, C. T., Ko, L. W., Chung, I. F., Huang, T. Y., Chen, Y. C., Jung, T. P., and Liang, S. F. (2006). Adaptive EEG-based alertness estimation system by using ICA-based fuzzy neural networks. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 53(11), 2469-2476. <https://doi.org/10.1109/TCSI.2006.884408>
- [3] Brandt, T., Stemmer, R., and Rakotonirainy, A. (2004, October). Affordable visual driver monitoring system for fatigue and monotony. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on* (Vol. 7, pp. 6451-6456). IEEE.
- [4] Rau, P. S. (2005, June). Drowsy driver detection and warning system for commercial vehicle drivers: field operational test design, data analyses, and progress. In *19th International Conference on Enhanced Safety of Vehicles* (pp. 6-9).
- [5] Von Jan, T., Karnahl, T., Seifert, K., Hilgenstock, J., and Zobel, R. (2005, June). Don't sleep and drive-VW's fatigue detection technology. In *Proceedings of 19th International Conference on Enhanced Safety of Vehicles*, Washington, DC.
- [6] Cai, H., & Lin, Y. (2007). An experiment to non-intrusively collect physiological parameters towards driver state detection (No. 2007-01-0403). *SAE Technical Paper*.
- [7] Nakagawa, T., Kawachi, T., Arimitsu, S., Kanno, M., Sasaki, K., and Hosaka, H. (2006). Drowsiness detection using spectrum analysis of eye movement and effective stimuli to keep driver awake. *DENSO Technical Review*, 12(1), 113-119.
- [8] Le, H., Dang, T., and Liu, F. (2013, December). Eye blink detection for smart glasses. In *Multimedia (ISM), 2013 IEEE International Symposium on* (pp. 305-308). IEEE.
- [9] Chen, Y. W., and Kubo, K. (2007, November). A robust eye detection and tracking technique using gabor filters. In *Intelligent Information Hiding and Multimedia Signal Processing, 2007. IHHMSP 2007. Third International Conference on* (Vol. 1, pp. 109-112). IEEE.
- [10] Jung, S. U., and Yoo, J. H. (2006, December). Robust eye detection using self quotient image. In *Intelligent Signal Processing and Communications, 2006. ISPACS'06. International Symposium on* (pp. 263-266). IEEE.
- [11] Abu-Faraj, Z. O., Mashaalany, M. J., Sleiman, H. C. B., Heneine, J. L. D., & Al Katergi, W. M. (2006, August). Design and development of a low-cost eye tracking system for the rehabilitation of the completely locked-in patient. In *Engineering in Medicine and Biology*

- Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE (pp. 4905-4908). IEEE.
- [12] Raudonis, V., Simutis, R., & Narvydas, G. (2009, November). Discrete eye tracking for medical applications. In *Applied Sciences in Biomedical and Communication Technologies, 2009. ISABEL 2009. 2nd International Symposium on* (pp. 1-6). IEEE.
- [13] Fritz, G., and Paletta, L. (2010, September). Semantic analysis of human visual attention in mobile eye tracking applications. In *Image Processing (ICIP), 2010 17th IEEE International Conference on* (pp. 4565-4568). IEEE.
- [14] Hassan, N. M., and Mansor, W. (2014, March). Detection of eye movements for controlling a television. In *Signal Processing & its Applications (CSPA), 2014 IEEE 10th International Colloquium on* (pp. 257-260). IEEE.
- [15] Mehrubeoglu, M., Pham, L. M., Le, H. T., Muddu, R., and Ryu, D. (2011, October). Real-time eye tracking using a smart camera. In *Applied Imagery Pattern Recognition Workshop (AIPR), 2011 IEEE* (pp. 1-7). IEEE.

7 Authors

Worawut Yimyam received the B.E. degree in Computer engineering from the Rajamangala University of Technology Thanyaburi, Thailand, in 2000, and the M.E. degree in Information Technology from the University of Rangsit, Thailand, in 2007 and the study Ph.D. degrees in Information Technology from Information Technology, King Mongkut's University of Technology North Bangkok, Thailand. He has held instructor positions at faculty of business administration from Rajamangala University of technology thanyaburi, Thailand, in 2001. He is currently a Lecturer at Faculty management science from Rajabhat phetchaburi, Thailand (e-mail: s5607011956056@email.kmutnb.ac.th).

Mahasak ketcham received the B.B.A. in Business Computer Siam University, Thailand, MS.I.Ed. Computer Technology, King Mongkut's University of Technology North Bangkok, Thailand and Ph.D. in Computer Engineering, Chulalongkorn University, Thailand. Dr. Mahasak ketcham is an assistant professor in Department of , Information Technology Management, King Mongkut's University of Technology North Bangkok, Thailand (e-mail: mahasak.k@it.kmutnb.ac.th).

Article submitted 28 July 2017. Resubmitted 05 November 2017, 16 July 2018 and 14 September 2018. Final acceptance 14 September 2018. Final version published as submitted by the authors.