International Journal of Technology (2018) 4: 819-827
ISSN 2086-9614

IMPLEMENTATION OF AN ON-BOARD EMBEDDED SYSTEM FOR MONITORING DROWSINESS IN AUTOMOBILE DRIVERS

Ejidokun T.O.[1*], Yesufu T.K.[2], Ayodele K.P.[2], Ogunseye A.A.[2]

[1]*Department of Computer Engineering, Afe Babalola University, Ado-ekiti, 36001, Nigeria*
[2] *Department of Electronic and Electrical Engineering, Obafemi Awolowo University, Ile-Ife, Nigeria*

(Received: Februari 2018 / Revised: March 2018 / Accepted: April 2018)

ABSTRACT

The development of technologies for detecting or preventing drowsiness at the wheel has been a major challenge in the area of accident avoidance systems. Due to the hazard that drowsiness presents on the road, methods need to be developed for its early detection. This study implements a Haar cascade technique on a Raspberry Pi module and evaluates the performance of the developed system. The results obtained from the evaluation of the standalone embedded system show that a precision of 80.11% and recall (sensitivity) of 99.81% were achieved. The results of the system usability test (based on an administered questionnaire) reveal that the mean System Usability Scale (SUS) score for the 20 participants is 77.38, with a standard deviation of 9.40. The minimum and maximum score are 57.50 and 92.50, respectively. The mean SUS score of 77.38 indicates that user satisfaction is adequate.

Keywords: Computer vision; Drowsiness; Haar cascade; Non-invasive; Raspberry Pi

1. INTRODUCTION

Thousands of people are killed or seriously injured every year due to drivers falling asleep at the wheel. Recent studies show that drivers' drowsiness accounts for up to 20% of accidents on motorways and monotonous roads, by impairing drivers' judgment and perception and their ability to control their vehicles (Anjali et al., 2016). In Nigeria alone, it is estimated that over 90% of road traffic accidents (RTAs) are due to driver error (Adekoya et al., 2011). The multiple factors contributing to the risk of RTA among long distance truck drivers in Nigeria include daytime sleepiness, fatigue and stimulant use (Atubi, 2012) Furthermore, in an American poll carried out by National Sleep Foundation in 2005, 60% of adult drivers (168 million people) admitted to having felt drowsy at the wheel; 37% (103 million people) had actually fallen asleep at the wheel; while 4% (approximately 11 million people) admitted to have had an accident due to fatigue whilst driving (National Sleep Foundation, 2011). These statistics show that safe driving is a major concern to societies all around the world. A considerable number of crashes have been attributed to drowsiness and fatigue; the World Health Organization (WHO) has predicted that traffic accidents will become the fifth leading cause of death by 2030 (WHO, 2009).

In recent years, drowsiness detection systems have become a keen topic of interest in both academic and industrial automobile communities, due to their potential to reduce fatalities caused by road accidents. Addressing the need for a reduction in crashes related to driver

---

*Corresponding author's email: engrtayo@gmail.com, Tel. +2348062399758
Permalink/DOI: https://doi.org/10.14716/ijtech.v9i4.1691

drowsiness will require innovative concepts and evolving methodologies (Barr et al., 2005; Suryanegara et. al., 2015). In-vehicle technological approaches, both already available and emerging, which use an on-board device that monitors the driver’s state in real time, may have real value as a safety net (Shinde & Mane, 2015). For this reason, it is believed that driver drowsiness can be effectively managed, thereby resulting in a significant reduction in related risk and improved safety (Ejidokun et al., 2011).

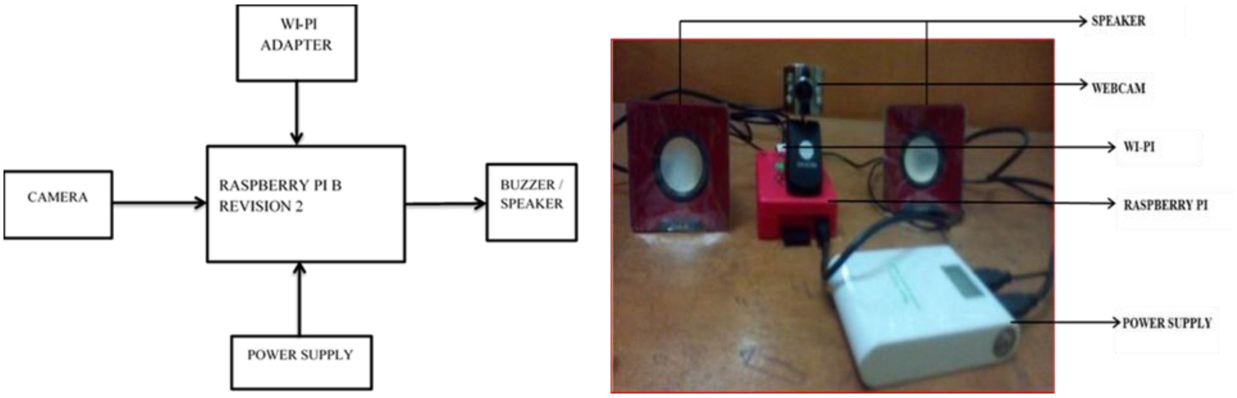
The aim of the study is to develop a cost-effective, non-intrusive system for monitoring the state of alertness of an automobile driver in real time and evaluate the performance of the developed system in terms of its effectiveness, efficiency and usability. The paper is organized as follows: Section 2 discusses the major hardware components of the developed system. Device setup, configuration for remote access and monitoring are also examined. Furthermore, a detailed explanation of the installation and testing of the device in an automobile is presented. The results obtained are presented and discussed in Section 3.

2. METHODS

2.1. Hardware Setup

The hardware setup consists of a Raspberry Pi module, a Wi-Pi adapter, webcam, speaker and power supply. Figure 1a shows a block diagram of the components of the system. A Raspberry Pi B+ revision 2 was used for the implementation. This consists of an ARM 1176JZF-S processor running at 700 MHz clock speed; a Video Core (IV) Graphic Processor Unit (GPU); 512 MB SDRAM shared with the GPU; two Universal Serial Bus (USB) ports; one video and audio output port; one 100 Mbit/s Ethernet port; and one high-definition multimedia interface (HDMI) output. It also has 26 pins, including eight general purpose input/output (GPIO) pins, one I^2C® bus, one SPI® bus, one UART bus and 3.3V, 5V and ground (GND) power supply pins. The Raspberry Pi does not have an on-board chip memory; it uses an external Secure Digital (SD) card to run its operating system and to store user data. This can be accessed either by connecting a keyboard, mouse and monitor, or be remotely monitored via a Secure Shell (SSH) terminal on a local area network.

A Lightwave LG 700 webcam was used for video acquisition, comprising a colour Complementary Metal Oxide Semiconductor (CMOS) image sensor with a resolution of up to 640×480 pixels and a frame rate of up to 30 frames/sec. The Wi-Pi is a 802.11n compliant wireless client, operating at a speed of up to 150 bps, with a frequency of up to 2.5 GHz and a transmitting power of 20 dBw.



(a) (b)

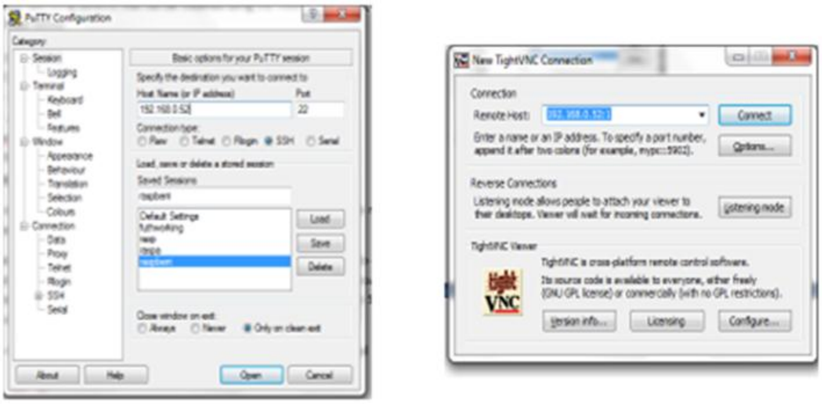
Figure 1 (a) Block diagram of drowsiness system; and (b) Physical setup of the system

A Havit stereo computer speaker, powered by a 5V direct current source, with an output power of 4w, was used to produce an output signal. Figure 1b shows the physical setup of the system.

2.2. Device Setup and Remote Access Configuration

The Raspberry Pi module was enclosed in a protective casing after it had been unpacked, so as to make the device handy and durable. The module needs to run on an operating system (OS); in order to install this, Raspbian setup was downloaded from the product's official website (Raspberry Pi F., 2017). A win32disk imager was used to write the OS image onto the SD card using a personal computer; subsequently, the SD card was inserted into the Pi, which was connected to a power supply. When the status LED began to blink, this indicated that the system had been successfully installed.

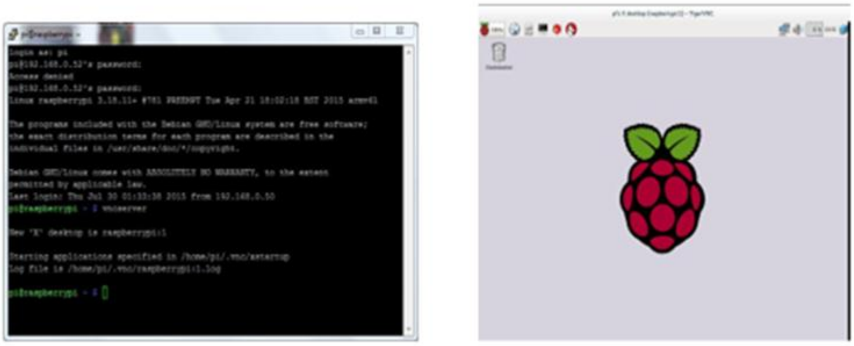
The system was controlled and monitored remotely using Putty and Virtual Network Computing (VNC) client software. Putty was used to establish a connection via the secure shell (SSH) terminal between the Pi and the computer, while the VNC client was used to remotely view and control the device using the remote frame buffer protocol to transmit keyboard and mouse events and to relay the graphical screen update over the network to the computer interface. Figures 2b and c show the configured interface using Putty and the VNC client.



Host Name (or IP address): 192.168.0.52
Port: 22
Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial
Close window on exit: ○ Always ○ Never ● Only on clean exit

(a) Putty configuration

(b) VNC viewer interface



(c) Logging into the Raspberry Pi

(d) Raspberry Pi desktop

Figure 2 Configuration and setting up of the Raspberry Pi

IP addresses of 192.168.0 52, port 22, and 192.168.0.50, port 5901 were assigned to Putty and the VNC client respectively. A username and password are required to log onto the device; the default user name and password used were "Pi" and "Raspberry", respectively, as shown in Figures 2c and d. Figure 3 shows the system setup. The captured image from the webcam is processed by the Raspberry Pi module, monitored and controlled remotely on a PC via a wireless network.

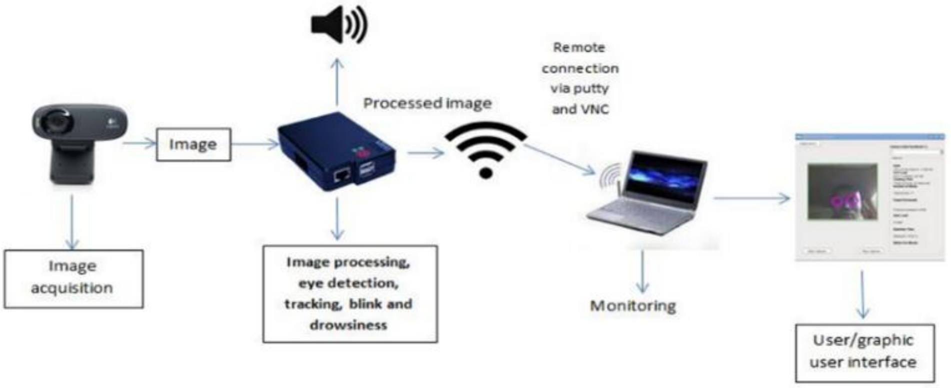


Figure 3 Remote control and monitoring over a wireless network

2.3. Algorithm Implementation

The drowsiness detection algorithm was implemented in C++ language using the Open Computer Vision (OpenCV) Library. C++ is the preferred software development language; it is known to be the closest language to the hardware interface (also referred to as a native language to the kernel) because it gives low level access to the hardware and memory for easy manipulation of operations in achieving the desired goal. It also has higher execution speed. The QtCreator Integrated Development Environment was used for the programing in C++; this allowed the combination of many APIs (application programming interfaces) to achieve the design and deployment of the drowsiness detection system. It provides user interface widgets in its toolbox, which facilitate quick prototyping of applications on the Raspberry Pi. OpenCV® is an open source Computer Vision project that aims to provide a development platform for computer vision algorithms through its collection of libraries and applications (OpenCV, 2001). It provides a cross-platform middle-to-high level API with hundreds of image processing and computer vision C functions and C++ classes.

Figure 4 shows the flowchart of the developed system. On initialization, video acquisition of the driver’s face takes place with the aid of the camera. The camera consists of a sensor which converts the analog signal into (video frame) digital form. The received video frame, in digital form, is processed by the vision controller, which is designed using features of Haar classifiers for object detection. It classifies objects based on rectangle features (Haar-like features), which are represented by the sums of the pixel values in the rectangular areas (Kulkarni & Shinde, 2017). In other words, it identifies the basic features of the human eye and extracts them from the face. It also maintains the position of the detected location, with respect to the movement of the object whose features are being tracked. The system is designed to reinitialize at the instant a tracked object cannot be located; this is often caused by rapid head movements. Once the eye has been detected and tracked, blinks are detected by simultaneous estimation of the cycles of the opening and closing of the eye within a period of time known as blink duration.

Computation of the blink duration ($T_{e)}$ is given by:

$$T_e = t_1 - t_0 \tag{1}$$

where t_0 is the period of eye closure, and t_1 is the period in which the eye is open. The computed blink duration is compared with an established blink duration threshold. If the blink duration is greater than a threshold value of 0.5s (minimum valid blink duration), a detected blink is recorded as a blink; otherwise, it is considered as noise resulting in error detection and is discarded (Thorslund, 2003).

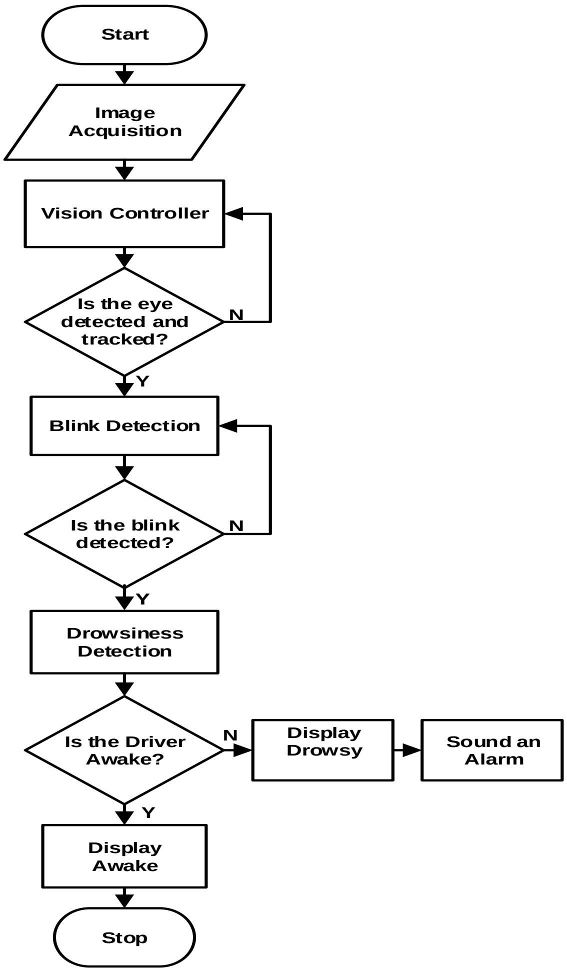


Figure 4 Flowchart of a drowsiness detection system

Subsequently, the average blink interval (*avbi*), blink duration range (*bdr*) and average blink duration (*avbd*) are calculated with Equations 2, 3 and 4, respectively.

$$avbi = \frac{\sum_{i==2}^{n} \mathit{start}\ \text{time of } \mathrm{b_i} - \mathit{start}\ \text{time of } \mathrm{b_{i-1}}}{n} \forall \mathrm{b_i} \in \mathrm{B} \tag{2}$$

$$bdr = \max(\mathit{duration\ of}\ b_i) - \min(\mathit{duration\ of}\ b_i) \forall b_i \in B \tag{3}$$

$$avbd = \frac{\sum_{i=1}^{n} \mathit{duration}\ \text{of } \mathrm{b_i} - \text{duration of } \mathrm{b_{i-1}}}{\mathrm{n}} \forall b_i \in B \tag{4}$$

$$B = \{b_{i.} \ldots\ldots\ldots\ldots b_n\} \tag{5}$$

where *B* is the blinks set, *n* is the number of blinks detected within time interval T_e, and b_i is the blink count for every detected blink.

Based on the computations in Equations 2, 3 and 4, the detected blinks are analyzed to produce various results, such that if the average blink duration is greater than the sleepy blink duration difference threshold, which is set to 3s, the system indicates that driver is feeling “sleepy”. In addition, whenever the average blink duration is greater than the drowsy blink duration threshold set to 0.8s, the system indicates that the driver is “drowsy”. The system is programmed to produce an audio alert via a recorded voice to alert a driver who is in a drowsy state. Hence, if the driver is neither “drowsy” nor “sleepy” the system concludes that he or she is likely to be awake. These threshold values produced the most accurate detection during the experimentation. Furthermore, the sleepy blink duration is largely based on the assumption that if people close their eyes for 3 or more seconds, they are likely to be "sleepy".

2.4. Device Installation and Testing

The Raspberry Pi module was installed on a Toyota Hiace Bus (2005 model) by fastening it in a suitable position, in such a way that it was not affected by vibration or shock. The camera was placed in a central position on the driver's dashboard, as shown in Figure 5, to enable it to capture the driver's face, irrespective of the change in eye position due to head movements. It was observed that, due to differences in height, test subjects have varying horizontal face levels to the dashboard.

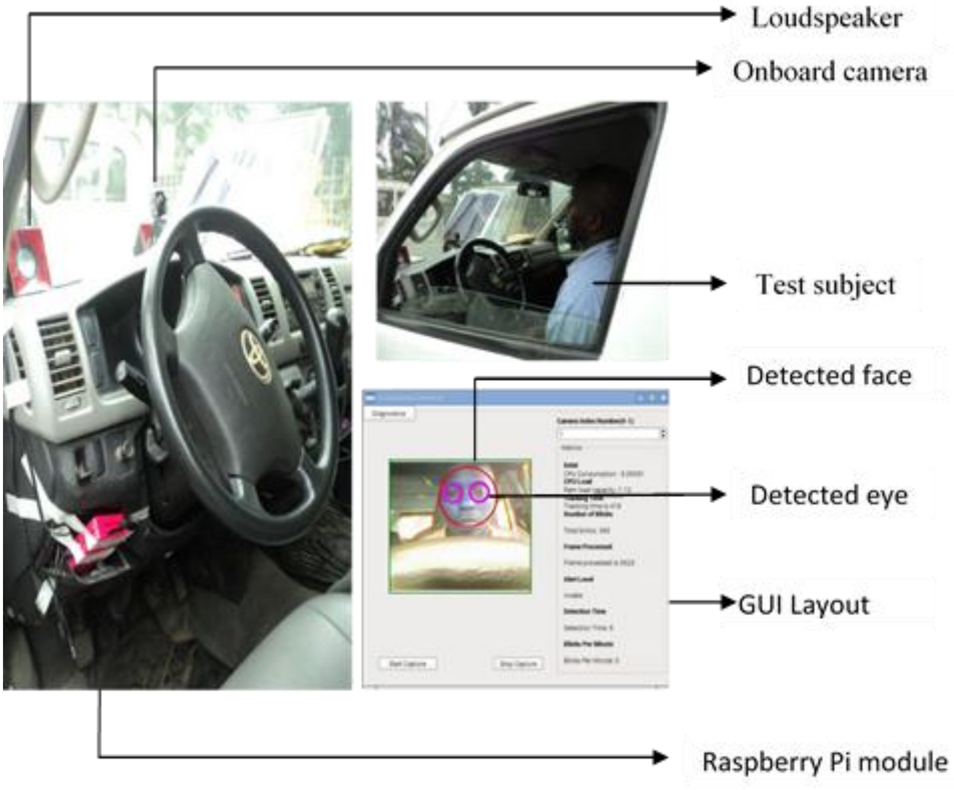


Figure 5 Installation, testing and output monitoring

The system was designed to carry out face detection in order to ascertain the presence of the driver at the wheel and to ensure that the camera was at a perfect viewing angle for accurate face capture. It acknowledges that the face has been detected through a programmed voice alert. Otherwise, its prompts the driver to adjust the camera to enable it to capture the face before the journey starts.

The system was powered by turning on the ignition; it took about 45–60 seconds for the system to boot up. The system was tested using 20 drivers, with four sessions per day between the hours of 8 am-10 am, 10 am-12 pm, 12 pm-2pm and 2pm-4pm over 5 days. The performance of the device was continually monitored remotely via a Wi-Pi adapter connected to the system.

3. RESULTS AND DISCUSSION

3.1. Precision and Recall

Precision is the measure of the consistency of a system in producing relevant results, while recall is the measure of its degree of sensitivity. Precision and recall in this case were calculated based on Equations 6 and 7.

$$\%Precision = \frac{TP}{TP+FP}\times 100 \quad (6)$$

$$\%Recall = \frac{TP}{TP+FN}\times 100 \quad (7)$$

where *TP* is True positive, *FP* is False positive, and *TN* is False negative.

True positive is the number of frames in which the “awake” state was detected, while False positive is the number of frames in which either a “sleepy” or “drowsy” state was detected (incorrect detection). False negative is the number of frames during which the system does not recognize any alert state, flagging “undefined”. It was assumed that the driver was initially in an “awake” state. Table 1 shows the precision and recall results. The experimental evaluation was conducted with 20 drivers for a duration of 7200 seconds (2hrs). Measurement was based on the number of frames captured and processed within this duration for each test subject. A precision of 80.11% and recall (sensitivity) of 99.81% were achieved. This is an indication that the degree of sensitivity and consistency in the system to produce relevant results is high.

Table 1 Estimation of the precision and recall of the developed system

Subject	TP	FP	FN	% Precision	% Recall
1	106916	43616	180	71.03	99.59
2	147031	2866	181	98.09	99.88
3	133943	16771	183	88.87	99.86
4	120583	30111	179	80.02	99.85
5	100962	45702	176	68.84	99.83
6	145383	3635	182	97.56	99.87
7	73455	48936	179	60.02	99.76
8	115833	34767	200	76.91	99.83
9	116795	28532	197	80.37	99.83
10	65558	82734	207	44.21	99.69
11	94585	48770	171	65.98	99.82
12	124943	19648	180	86.41	99.86
13	143200	4196	192	97.15	99.87
14	67299	85374	195	44.08	99.71
15	120548	3520	179	97.16	99.85
16	84622	67049	204	55.79	99.76
17	147294	4253	205	97.19	99.86
18	143547	4284	196	97.10	99.86
19	124896	2078	200	98.36	99.84
20	143572	4491	196	96.97	99.86
Total	2320965	581333	3782	80.11	99.81

3.2. System Usability Scale

The system usability scale (SUS) is a simple 10-item scale which gives a global view of the subjective assessments of usability (Brooke, 1996). The aim of this evaluation is to ascertain the effectiveness, efficiency and user satisfaction of the system. A questionnaire was distributed to the 20 respondents, who consisted of drivers within the age bracket of 30−45 years old. This was used to evaluate the developed system. In accordance with the interpretation of the SUS scores, products with scores of less than 50 are judged to be unacceptable; those with scores between 50−70 are marginally acceptable; and products with scores above 70 are passable (Bangor et al., 2008). The obtained results show that the mean SUS score for all participants is 77.38, with a standard deviation of 9.40. The minimum score is 57.50 and the maximum 92.50. Figure 6 also shows that five participants gave scores of between 50−70, while fifteen gave above 70. The mean SUS score of 77.38 obtained indicates that the system effectiveness, efficiency and user satisfaction is adequate.

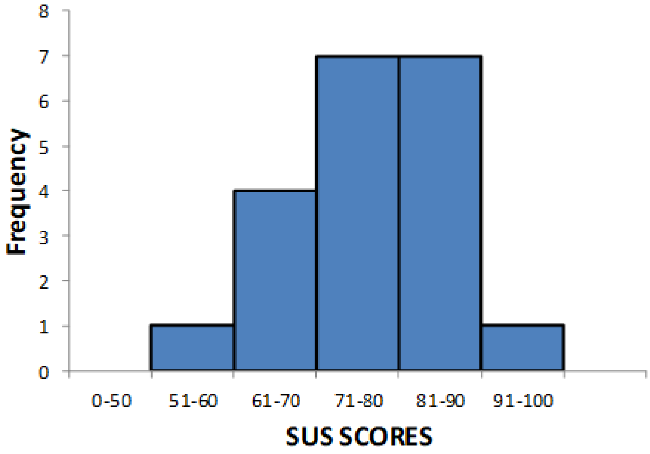


Figure 6 Histogram showing the frequency distribution of the computed SUS scores

4. CONCLUSION

The study has implemented a drowsiness detection system on a single board computer. Wireless network capability was added to enhance the continuous remote monitoring of the alertness status of the driver. The alertness level of the driver can be monitored offline inside the vehicle with the aid of an audio alert via a recorded voice. The low power consumption of the platform enables the seamless integration of the device into the vehicle power system without affecting its performance.

An on-road experiment was conducted to ascertain the effectiveness of the system in determining if a driver was in a normal state of mental alertness. The system shows a high degree of consistency and responsiveness in detecting a driver in an “awake state” and issues a warning note once the system detects otherwise, with minimum error. Furthermore, the short-term system usability evaluation carried out indicates that the test drivers found the device easily understandable, usable and intuitive, and were comfortable with its operation.

In further work, a car simulator will be used to collect physiological data for validation of the effectiveness of the developed system in detecting a driver in a drowsy state. Subsequently, long-term evaluation will be carried out to enable us to further refine the functionality of the system.

5. REFERENCES

Adekoya, B.J., Adekoya, A.O., Adepoju F.G., Owoeye, J.F.A., 2011. Driving under Influence among Long Distance Commercial Drivers in Ilorin, Nigeria. *International Journal Biology Medical Research*, Volume 2(4), pp. 870–873

Anjali, K.U., Thampi, A.K., Vijayaraman, A., Francis, M.F., James, N.J., Rajan, B.K., 2016. Real-time Nonintrusive Monitoring and Detection of Eye Blinking in View of Accident

Prevention due to Drowsiness. *In:* International Conference on Circuit, Power and Computing Technologies (ICCPCT), pp. 1–6

Atubi, A.O., 2012. Determinants of Road Traffic Accident Occurrences in Lagos State: Some Lessons for Nigeria. *International Journal of Humanities and Social* Science, Volume 2(6), pp. 252–259

Bangor, A., Kortum, P.T., Miller, J.T., 2008. The System Usability Scale (SUS): An Empirical Evaluation. *International Journal of Human-Computer Interaction*, Volume 24(6), pp. 574–594

Barr, L., Howarth, H., Popkin, S., Carroll, R.J., 2005. A Review and Evaluation of Emerging Driver Fatigue Detection Measures and Technologies. Available online at *http://www.ecse.rpi.edu/~qji/Fatigue/fatigue_report_dot.pdf.*, pp.1–27

Brooke, J., 1996. SUS-A Quick and Dirty Usability Scale. *Usability Evaluation in Industry*, Volume 189(194), pp. 4–7

Ejidokun, T.O., Ayodele, K.P., Yesufu, T.K., 2011. Development of an Eye-blink Detection System to Monitor the Drowsiness of Car Drivers. *Ife Journal of Technology*, Volume 20(2), pp. 51–55

Kulkarni, A.S., Shinde, S.B., 2017. A Review Paper on Monitoring Driver Distraction in Real Time using Computer Vision System. *In:* IEEE International Conference on Electrical, Instrumentation and Communication Engineering (ICEICE), pp. 1–4

National Sleep Foundation, 2011. Facts and Stats. Available online at http://drowsydriving.org/about/facts-and-stats/, Accessed 27th August, 2015

OpenCV., 2001. *Open Source Computer Vision Library: Reference Manual*. Intel Corporation, USA

Raspberry Pi F., 2017. Raspberry Pi Desktop (for PC and Mac). Available online at https://www.raspberrypi.org/downloads/, Accessed 24th May, 2017

Shinde, P.A., Mane, Y.B., 2015. Advanced Vehicle Monitoring and Tracking System based on Raspberry Pi. *In:* IEEE Ninth International Conference on Intelligent Systems and Control (ISCO), pp. 1–6

Suryanegara, M., Nugraha, I.G.D., Adhi, B.A., Lubis, M. F., Putra, M.R.E., 2015. The Local Innovation Perspective: Development of Mobile-Herbal Service for Indonesia's Mobile Cellular Market. *International Journal of Technology*, Volume *6*(2), pp. 109–120

Thorslund, B., 2003. Electrooculogram Analysis and Development of a System for Defining Stages of Drowsiness. *Published M.Sc. Thesis*, Linköping University, Linköping Sweden

WHO (World Health Organization), 2009. Global Status Report on Road Safety: Time for Action. Geneva