Smart Driver Monitoring System

BY

SASIKUMAR PERIYASAMY

Submission date: 15-May-2020 01:32PM (UTC+0530)
Submission ID: 1317371694
File name: Smart_Driver_Monitoring_System.docx (44.83M)
Word count: 3215
Character count: 17925

Smart Driver Monitoring System

Shubhi Shaily • Srikaran Krishnan • Saisriram Natarajan • Sasikumar Periyasamy

Abstract A countless number of people drive on the highway day and night. The nature of the job is such that physical and mental exertion is massive on the driver's body and this could lead to a slew of primary and secondary problems. Primary issues mainly include the driver and his wellbeing and the secondary issue which we deal with in this study is the rise of fatigue related accidents caused mainly due to negligence. An additional issue that bothers us is the unpredictability of the human body and optimal conditions and the associated errors caused thereby. There are no strict norms in place to handle such problems related to driver health and this in turn can place a great risk on human life and materialistic loss.

Through this novel python project of ours we plan to develop a driver drowsiness detector based on image processing. The following solution is developed with the aim to provide a novel solution to driver authentication and subsequent fatigue and drowsiness detection on-board whilst the driver is driving the car. The mechanism provided is both non-intrusive and involves the use of artificial intelligence that will provide accurate and desired results thereby avoiding damage generally caused due to negligence and imposters.

The system created will work based on vehicle details received from the OBD-II and the camera mounted on the dashboard to monitor the driver.

Keywords *Artificial Intelligence, OBD-II, Authentication, Ignition, raspberry pi, mobile app, on-board camera, fatigue, drowsiness, alert.*

Sasikumar Periyasamy
sasikumar.p@vit.ac.in

Shubhi Shaily
shubhishaily2016@vitstudent.ac.in

Srikaran Krishnan
k.srikaran2016@vitstudent.ac.in

Saisriram Natarajan
saisriramnatarajan2016@vitstudent.ac.in
Department of Communication Engineering, School of Electronics Engineering, VIT University,
Vellore, India

1 Introduction

Fleet management is an umbrella term which encompasses the broad solution list used by the industry in reference to managing the fleet of commercial vehicles usually handled by the likes of ride sharing companies and so on. The intricacies involved in managing a huge fleet of automobiles are diverse and complex at the same time. The fleet manager is responsible to undertake any responsibility in case an unfortunate incident might occur and the driver is a victim there. Proper Procedure with regards to insurance related mechanisms needs to be established to enable drivers to deal with an accident such that claims can be handled quickly and efficiently which will allow the necessary repair work to be carried out. Hence organizations work to cope up with such problems in a segmented way such that they can keep with such vehicle related issues.

The major discrepancy faced by such organizations is in the management of its inventory – the cars and their respective human capital in the form of drivers. With the advent of Internet of things and connected technology, a major gap in solving such peculiar problems of a remote and dynamic nature are solved with least human effort and faster response times. Incorporation of such a system does take a huge step in maintaining pedestrian safety and is a huge benefactor for such fleet management organizations.

The common issue that is solved is that of driver related drowsiness and fatigue that is associated with long hours of driving. The on-board camera monitors specific features of the driver's face [1][9] and works upon an algorithm [9] upon which it decides if the driver is drowsy or not. If the driver is found drowsy, an alert is passed to the mobile app using the raspberry pi in addition to playing a loud alert noise in the car that gets the attention of the driver.

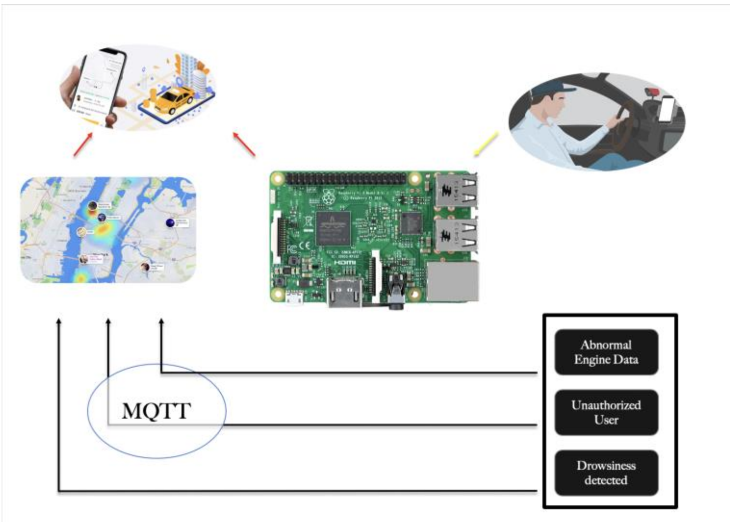

Fig 1: Network diagram of the system

2 System Structure and Modules

The system is an ecosystem with the Raspberry Pi mainly holding every procedure together. The cycle starts from the Vehicle where data pertinent to its health is acquired continuously and sent to the raspberry pi which then sends the data to the Android Mobile application as a driver data profile. At the same time the driver is monitored through an on-board camera that detects the driver constantly for signs of drowsiness and fatigue and makes sure the driver is not losing attention while on the road. This is done through an Artificial neural network model [1][10] to detect the face and a subsequent mathematical model that checks for drowsiness. These image frames are constantly sent to the Raspberry Pi which runs an algorithm [10] to check for drowsiness and in case such susceptible markers are identified, the notification is sent to the app as well as an alarm goes off in the car to alert the driver.

2.1 Hardware Modules

2.1.1 Raspberry Pi 3

The Raspberry pi 3 Model B [13] is a small-sized computer with a Linux based operating system known as Raspbian. It is a powerful and handy tool as it consists of 1 GB RAM, a DSI display port, a USB port, BCM43143 on board Wi-Fi with forty pins extended GPIO. Interfacing of various modules is done through the GPIO pins. The micro SD port is used for loading operating system and storing data.

Fig 2: Raspberry Pi

It is battery efficient as it only needs 5V of battery. Various libraries can be installed on it as it is a Linux based machine, leading to more convenient instructions to the computer.

In the beginning, during its setting up time it must be updated with the latest packages [13].

2.1.2 Raspberry Pi Camera

This camera sensor is a custom designed add-on board for raspberry pi with a high quality 8 megapixel SONY IMX219 image sensor. It is fit for catching 3280 x 2464-pixel static pictures and supports 1080p30, 720p60 and 640x480p60/90 video spilling and recording offices. Its minimal size and light (a little more than 3 g), make it reasonable for compact applications. It tends to be associated with Raspberry Pi through a short lace link.

.

Fig 3: Pi Camera

2.1.3 Rs232 to TTL Converter

This module is utilized for the transformation of the RS232 voltage of PC to TTL voltage which is utilized by the microcontroller. Fitting the DB9 connector to the PC to speak with the microcontrollers utilizing the Serial port. It is pertinent to an assortment of stages including Arduino/AVR/ARM. For Arduino, a previously settled UART2USB association is required, with this single module you can accomplish increasingly sequential correspondence to your PC.

Fig 4: Rs232 to TTL Converter

3.0 Image processing and Drowsiness detection

What lies at the heart of this system is a novel solution to drowsiness aversion and the crux of this mechanism is image processing and facial recognition. The cyclic process monitors the face of the driver through various metrics and characteristics such that any particular offset indicating drowsiness is detected and an alert is played out aloud in the car and a notification is sent to the holding company through the android application.

3.0.1 HAAR Cascade

The image processing system is mainly concerned with facial recognition and the HAAR cascade is a quick and efficient Object detection system deployed here. The object to be detected here is the face of the driver.
HAAR Cascade is an object detection algorithm based on machine learning which is used in identifying objects in an image or video. Through a machine learning based approach, a cascade function is trained from a lot of images both positive and negative. It is then used for detection of objects in other images.
It is a robust and extremely rapid object detection algorithm developed in keeping facial recognition as a major motive which is inspired by the concept of features proposed by Paul Viola and Michael Jones [11]. This is extremely helpful in the situation presented here where frontal-facial detection should be done with equal parts reliability and rapid processing as possible outcomes depend highly on identification of drowsiness and the possible aversion of any mishap.

There are four stages to the algorithm:
HAAR Feature Selection, Creating Integral Images, Adaboost Training and Cascading Classifiers.

The process starts with the detection of the face and its characteristic features. This is achieved through HAAR Features that quickly identifies the respective feature through rectangular boxes as shown below. The value is calculated based on the number of pixel values inside the darker rectangle rather than the pixel value present [11] [14]. A HAAR feature considers neighbouring rectangular regions at a particular location in a detection window, performs a summation of pixel intensities and finally calculates the difference between these sums.
This HAAR like feature is instrumental in providing specific indication regarding the face (of the driver) to an image such that the process is rapid.

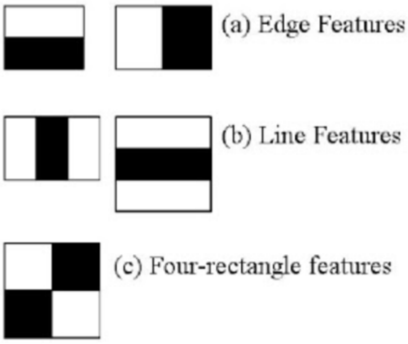

Fig 5: HAAR Feature Boxes

The Integral images are useful in faster computation of the object detection value which acts as a major quantitative defining factor to detect the object.
Adaboost selects the features that are the best and then trains the classifiers that use them [11] [15]. A strong classifier is constructed as linear combination of weighted simple weak classifiers [15].
A window the size of the target is superimposed over the input image, and also over different section the image as shown in the picture below. The HAAR features are then calculated. Non-objects are separated from object by comparing the difference to a learned threshold. Since each HAAR feature is just a "weak classifier", a huge number of HAAR features are required to describe an object with optimal accuracy and are hence are arranged into cascade classifiers with the aim to form a strong classifier [15].

Fig 6: HAAR Feature Boxes used to detect facial landmarks – shadow of an eye

The cascade classifier comprises of a variety of stages, wherein each stage is a collection of weak learners [11]. Simple classifiers are weak learners which are called Decision Stumps. Boosting technique is used at every stage. Through the boosting technique, by taking a weighted average of the decisions made an accurate classifier is built.

The present area of the sliding window is characterised as either positive or negative at every phase of the classifier. An object was found is indicated by positive whereas when no object is found is indicated by negative.

In the case that mark comes out to be negative, the order of this district is finished. It is then that locater slides the window to the next area.

The classifier is passed to the next stage if the indicator is positive. It is only when the last stage characterizes the district as positive that the indicator reports an item to be found at the present window.

3.0.2 Eye Aspect Ratio

Now that the face of the driver is detected using the HAAR Algorithm, we proceed onto detecting drowsiness which forms the heart of this research. This is done using a unique mathematical model and ration called the Eye Aspect Ratio [9]. The level of drowsiness of a driver is arrived at based on a characteristic called eye blink rate. A scalar value is arrived at to quantitatively define drowsiness which is calculated by the Eye Aspect Ratio (EAR) Formula given by the equation (1). For instance, if the driver's eye is closed beyond a certain time period or does blink very frequently, it indicates a level of drowsiness to be detected. Thus, it becomes pertinent to calculate eye blink rate in order to arrive at a conclusion regarding the state of the driver. The landmarks detected in each video frame is applied to the EAR formula such that the eye openness state is determined.

$$EAR = \frac{||p2-p6||+||p3-p5||}{2||p1-p4||} \qquad (1)$$

The landmarks are usually scalar coordinates between the length and width of the eye from each video frame of the driver. As shown in equation (1) the landmarks of the eye are defined using scalar coordinates from p1 to p6. The coordinates p2, p3, p5 and p6 help calculate the width of the eye whereas the coordinates p1 and p4 help calculate the length of the eye as shown in the figure below.

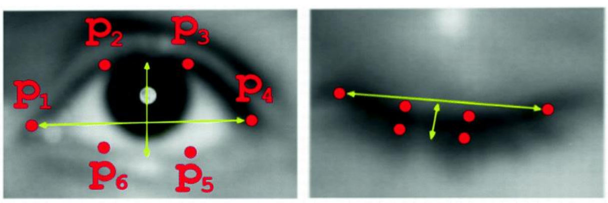

Fig 7: Scalar landmarks on the face used to find EAR value

The eye aspect ratio is a fixed value when the eye is opened, but quickly falls close to 0 when the yes a are in a closed state. The eye aspect ratio range for when eyes are open and close is shown by the

equation below. When the eyes are in a closed state, the eye aspect ratio value would be closer to 0, whereas when the eye is open, the EAR value can be an integer value greater than 0.

$$\text{EAR} = \begin{cases} x > 0; \text{eyes open} \\ 0; \ \text{eyes close} \end{cases} \tag{2}$$

3.1 On-board diagnostics

The On-board diagnostics standard connector is an on-board PC framework that is liable for checking your vehicle's engine, transmission, and emissions control segments. By and by, a greater part of street vehicles utilize OBD-II, permitting them to peruse the CAN transport information and bus data directly from vehicles. It works on a request/response basis.
Through the OBD we gather and store information about the vehicle location, vehicle velocity, engine revolutions, engine coolant temperature, and throttle position [16]. The OBD-II Bluetooth connector brings information straightforwardly from the vehicle, permitting the Pi to get information remotely [16].

pyOBD (otherwise known as pyOBD-II or pyOBD2) is a Python-based open-source OBD-II (SAE-J1979) agreeable sweep instrument programming. It interfaces with ease ELM 32x OBD-II indicative interfaces, for example, ELM-USB. It permits correspondence with the vehicle's ECU, shows shortcoming codes, shows estimated values, reads status tests, and so on. The program will associate through the OBD-II interface, show the measures accessible subject to the specific vehicle, and show constant motor information to the vehicle's secondary selling head unit in an intelligent GUI.

3.1.1 Software Installation

A working install of Raspbian with the network access is required before beginning. This is done using command line instructions.
Another software that needs to be downloaded is the OBD-Pi Software [12].

3.2 Data Transfer

The Raspberry Pi communicates with Android App via MQTT Protocol. Through this it sends the on-board diagnostics data if it is above the specified threshold limits. It also sends a notification in case drowsiness is detected.

MQTT brings several benefits including scalability, lightweight, security, and a reduction in energy consumption, essential for wireless communication. Computation takes place at the edge, thereby the data is only transferred only when abnormalities in data are detected, making it more cost and energy efficient.

3.3 User Interface and control(Android App)

The users are notified about the anomaly in the location, speed, throttle position, coolant temperature and rpm of the car via an android app. They are also made aware in case drowsiness is detected.

Each subscriber has a separate account, containing information about their entire fleet. This ensures privacy and security with data from end to end. Using the information on the app, the user can take necessary steps to prevent damage or misuse of the vehicle. Qualities of the driver as well as the vehicle can be assessed, allowing to effectively manage the fleet.

4 Implementation and Result

The figures 8 gives us an accurate run down of the system when it is up and running. It was observed that the end to end running time from facial recognition to detecting drowsiness, was optimized from the earlier time of 1.8s to 0.8s with the help of certain tweaks made to OpenCV Library.

Also, this system is advantageous due to the live stream of the driver provided such that any mishap due to negligence is averted. This way the system is non-intrusive as it doesn’t affect or compromise driver comfort in any way. Figures 9 and 10 is a working representation of the on-board data that is retrieved from the car and sent to the mobile app at the user side depicted by figure 11 and 12.

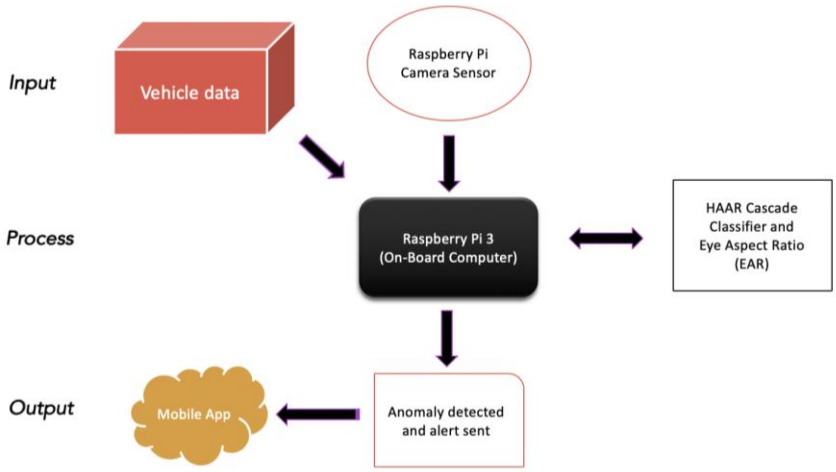

Fig 8: Network diagram of the system

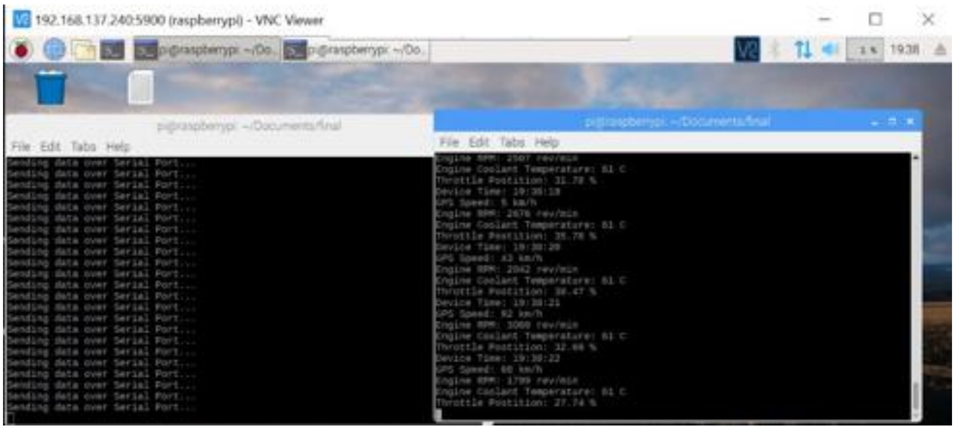

Fig 9: On-Board data

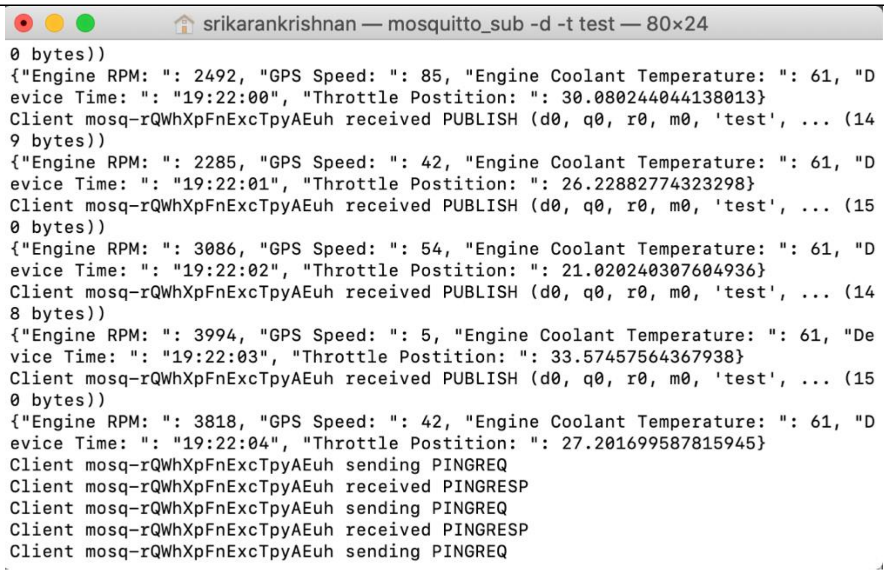

Fig 10: On-Board Data retrieved real time from the car

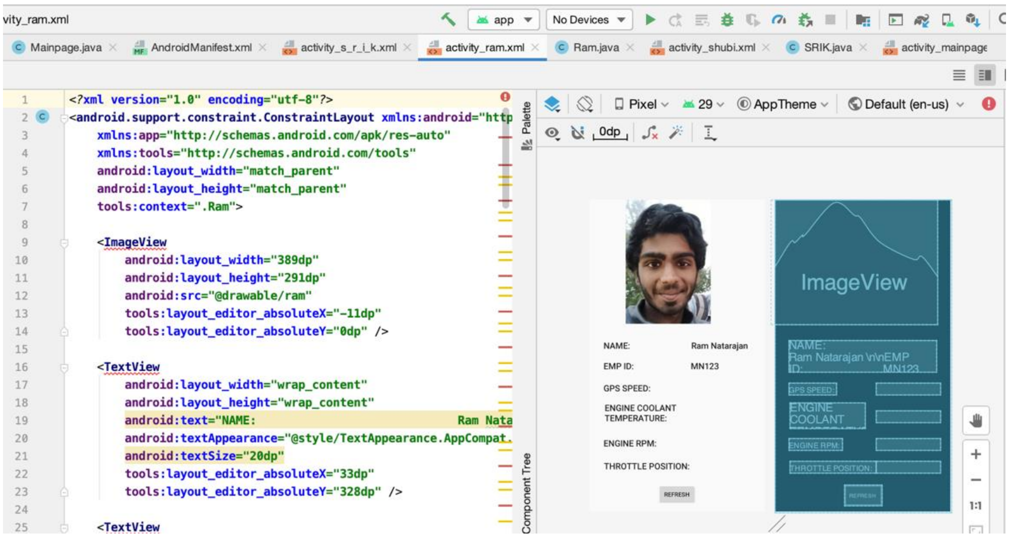

Fig 11: Android app displaying driver specific details and tracking

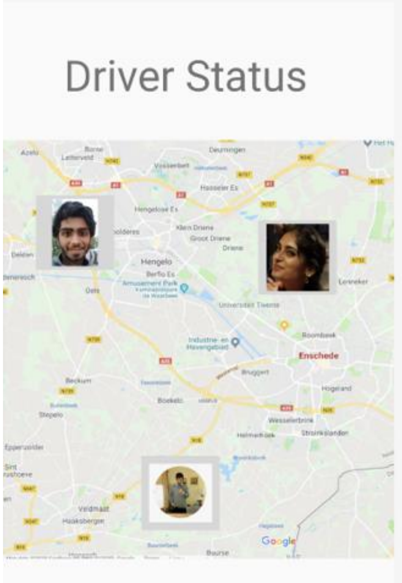

Fig 12: Android app displaying driver profiles

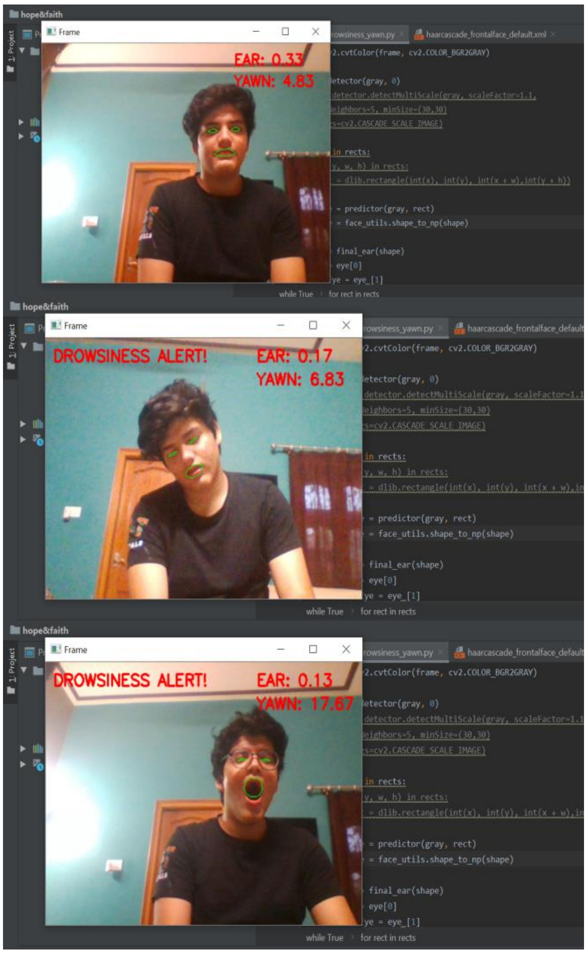

Fig 13: Camera displaying different states of driver

5 Conclusion and Future Work

Real-time data from the car is constantly monitored and the dash camera feed checks for parameters related to driver drowsiness and sounds an alarm successfully. The on-board vehicle data is displayed when required by the fleet management company as and when required to check any diversion or discrepancy. This system if imparted in large scale with robust fail checks would prove to be a boon for large fleet management companies and could infiltrate the private and public vehicle owned space effectively and efficiently. An added advantage is when the insurance company is involved where specific proof can be provided to avoid misuse or rather claim what is right.

This system could be developed further with a sensor-based system aiding the on-board dash cam in case certain physical features of the driver or certain circumstances where the driver footage isn't just enough to determine any anomaly. This way this novel work which addressed a culmination of different problem statements could be a great model for large logistics companies to further improve their control of inventory and goods.

6 References

[1] T. Vesselenyi, S. Moca, A. Rus, T. Mitran and B. Tătaru, "Driver drowsiness detection using ANN image processing" in IOP Conference Series: Materials Science and Engineering 252 (2017) 012097.

[2] Mingqi Lu, Yaocong Hu and Xiaobo Lu, "Driver Detection Based on Deep Learning" in 3rd Annual International Conference on Information System and Artificial Intelligence (ISAI2018) 22–24 June 2018, Suzhou, IOP Conf. Series: Journal of Physics: Conf. Series 1069 (2018) 012118.

[3] V Triyanti and H Iridiastadi, "Challenges in detecting drowsiness based on driver's behaviour" in Sustainable Development in Industry and Management, IOP Conf. Series: Materials Science and Engineering 277 (2017) 012042.

[4] John G. Gaspar1, Timothy L. Brown1, Chris W. Schwarz1, John D. Lee2, Julie Kang3, and James S. Higgins, "Evaluating Driver Drowsiness Countermeasures", Taylor & Francis.

[5] Zeimpekis, V.S., Tarantilis, C.D., Giaglis, G.M., Minis, I.E., Dynamic Fleet Management Concepts, Systems, Algorithms & Case Studies, Springer.

[6] Ying LT. Kanade T, Cohn JF. 2001, "Recognizing Action Units for facial expression analysis" in IEEE Trans Pattern Anal Mach Intell., 23(2) 97–115.

[7] Shamian M, Khan I, Abdullah H 2014, "Efficient drowsiness detection by facial features monitor" in Research J. Applied Sciences, Engineering and Technology 7 (11): 2376-2380.

[8] Lim S M and Chia S E 2015, "The prevalence of fatigue and associated health and safety risk factors among taxi drivers in Singapore" in Singapore Med J. 56 (2): 92–97

[9] Nora Kamarudin, Nur Anida Jumadi, Ng Li Mun, Ng Chun Keat, Audrey Huong Kah Ching, Wan Mahani Hafizah Wan Mahmud, Marlia Morsin, Farhanahani Mahmud, "Implementation of Haar Cascade Classifier and Eye Aspect Ratio for Driver Drowsiness Detection Using Raspberry Pi" in Universal Journal of Electrical and Electronic Engineering 6(5B)

[10] Caio B. Souto Maior , Márcio C. Moura, João M. M. de Santana , Lucas M. do Nascimento , July B. Macedo, Isis D. Lins and Enrique L. Droguett, "Real-time SVM Classification for Drowsiness Detection Using Eye Aspect Ratio" in Proceedings of Probabilistic Safety Assessment and Management PSAM 14, Springer: September 2018

[11] Paul Viola and Michael Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features" in Conference on Computer Vision and Pattern Recognition 2001

[12] https://github.com/peterh/pyobd

[13] Raspberry Pi Setup. https://www.raspberrypi.org. Accessed 27 Dec 2016.

[14] "Computational Vision and Bio Inspired Computing", Springer Science and Business Media LLC,2018

[15] Eman Alajrami, Hani Tabash, Yassir Singer, M.-T. El Astal. "On using AI-Based Human Identification in Improving Surveillance System Efficiency", 2019 International Conference on Electronic Technologies (ICPET), 2019

[16] Shi-Huang Chen, Jun-Yu Chen, Kuo-Yuan Lu. "The Use of Cloud Speech Recognition Technology in Vehicle Diagnosis Applications", 2011 Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, 2011

Shubhi Shaily is a final year student of B.Tech.(ECE with spec in IOT & Sensors) at Vellore Institute of Technology. Her area of interests comprises of: Internet of things, Machine Learning and Embedded systems

Srikaran Krishnan is a final year student of B.Tech.(ECE with spec in IOT & Sensors) at Vellore Institute of Technology. His area of interests comprises of: Internet of things, Machine Learning and Embedded systems

Saisriram Natarajan is a final year student of B.Tech.(ECE with spec in IOT & Sensors) at Vellore Institute of Technology. His area of interests comprises of: Internet of things, Machine Learning and Embedded systems

Sasikumar Periyasamy is an Assistant Professor (Senior) in VIT University. Awarded Ph.D. in QOS Enhancement For Distributed Clustering Techniques In Wireless Sensor Networks, VIT University, M.E. Applied Electronics from Government College of Technology (Autonomous Institution) and B.E. Electrical and Electronics from Anna University. He is an expert in the areas of Embedded Systems, Wireless Sensor Networks, Communication Engineering, and Internet of Things.