5 cm

CONNECTED DEVICES IN INTERNET-OF-THINGS (IOT) FOR DRIVER FATIGUE ALERT SYSTEM

7 cm

WAN MARDHIAH BT WAN MOHD YUSOF

8 cm

ELECTRICAL AND ELECTRONICS ENGINEERING
UNIVERSITI TEKNOLOGI PETRONAS
JAN 2017

Final Year Project 2 Final Report of

Connected Devices in Internet-of-things (IoT) for Driver Fatigue Alert System (16S041)

By

Wan Mardhiah bt Wan Mohd Yusof

(17403)

Electrical and Electronics Engineering

Supervised by

A.P Dr. Gunawan Witjaksono

Dissertation submitted in partial fulfilment of

requirements for the

Bachelor of Electrical and Electronics Engineering (Hons)

Submitted in May 2017

Universiti Teknologi PETRONAS,

Bandar Seri Iskandar,

32610 Seri Iskandar,

Perak Darul Ridzuan.

CERTIFICATION OF APPROVAL

Connected Devices in Internet-of-things (IoT) for Driver Fatigue Alert System (16S041)

By

Wan Mardhiah bt Wan Mohd Yusof

17403

A project dissertation submitted to the

Electrical and Electronics Engineering Program

Universiti Teknologi PETRONAS

In partial fulfilment of the requirement for the

BACHELOR OF ENGINEERING (Hons)

(ELECTICAL & ELECTRONICS)

Approved by,

____

A.P Dr. Gunawan Witjaksono

Universiti Teknologi PETRONAS,

Bandar Seri Iskandar,

32610 Seri Iskandar,

Perak Darul Ridzuan.

Jan 2017

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

____

WAN MARDHIAH BT WAN MOHD YUSOF

ABSTRACT

This report discusses about a driver fatigue alert system project related to connected devices in Internet-of-Things (IoT). It applies additional parameters and different device for detection process if compared to other researches that have been conducted. This project is aimed to help preventing accidents occurring due to tiredness or fatigue. The IoT comprises the combination of electronic device, sensor and network to work. It is applied to ensure the system can be implemented on any device, anyone, any service or business, any path or network, anywhere, and at any time or context. This system will detect fatigue condition from certain parameters, and give alert to notify if there are positive signs of fatigue. The system will also give suggestion to rest at the closest safe and suitable place to rest and reenergize. This system basically uses the application of mobile phone camera for detection and give alarm, and requires the presence of internet in the device to detect current location and suggest the closest location to stop accordingly. This systems uses facial extracted feature which includes: (1) mouth opening, (2) PERCLOS (Percentage of Eye Closure), and (3) head positioning. This project implements image processing method, or specifically Viola Janes Face Detection Algorithm to gather and process data. For programming part, the MATLAB® Computer Vision System command will be applied.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1.0

INTRODUCTION

1.1 Background of the Project

This project named as Connected Devices in Internet-of-things (IoT) for Driver Fatigue Alert System, is basically aimed to help preventing accidents occurring due to tiredness or drowsiness felt by drivers while on road. Fatigue felt by drivers can lead to losing focus and worse, falling asleep while driving. This situation may lead to accidents though the drivers are accidentally falling asleep for only few seconds.

IoT or Internet-of-Things comprises the combination of electronic device, sensor and also network to make it work. It can be implemented on any device, anyone, any service or business, any path or network, anywhere, and at any time or context. The systems are provided with identifier which is unique, and has the capability to transfer information or data through networking system. This IoT system is very ultimate because it communicates without a person communicate to a person, or even a person communicate to the device.

This system will help the drivers by detecting their fatigue condition from certain parameters, and give alert to drivers to notify if there are positive signs of fatigue. The system will also give suggestion for the drivers to rest at the closest safe and suitable place to rest and reenergize. This system basically uses the application of camera of mobile phone to detect the fatigue criteria of the driver and give alarm, and requires the presence of internet in the device to detect their current location and suggest the closest location to stop accordingly.

This systems will use facial extracted feature to obtain result either the driver is fatigue or not. The features are mouth opening, PERCLOS (Percentage of Eye Closure) and also head positioning. Thus, this project will implement image processing method, or specifically Viola Janes Face Detection Algorithm to gather and process the data. This technique can be applied to wide classes of figures or objects. Though, its application which is more suitable for recognizing facial features make it suitable to be implemented in this project. For programming part, the MATLAB® Computer Vision System command will be applied.

ABSTRACT

This report discusses about a driver fatigue alert system project related to connected devices in Internet-of-Things (IoT). It applies additional parameters and different device for detection process if compared to other researches that have been conducted. This project is aimed to help preventing accidents occurring due to tiredness or fatigue. The IoT comprises the combination of electronic device, sensor and network to work. It is applied to ensure the system can be implemented on any device, anyone, any service or business, any path or network, anywhere, and at any time or context. This system will detect fatigue condition from certain parameters, and give alert to notify if there are positive signs of fatigue. The system will also give suggestion to rest at the closest safe and suitable place to rest and reenergize. This system basically uses the application of mobile phone camera for detection and give alarm, and requires the presence of internet in the device to detect current location and suggest the closest location to stop accordingly. This systems uses facial extracted feature which includes: (1) mouth opening, (2) PERCLOS (Percentage of Eye Closure), and (3) head positioning. This project implements image processing method, or specifically Viola Janes Face Detection Algorithm to gather and process data. For programming part, the MATLAB® Computer Vision System command will be applied.

1.2 Problem Statement

There are few drowsiness or fatigue detection systems that currently being published. However, these monitoring systems demands the application of expensive equipment, complex computation process, and complex installation in the vehicle and might obstruct the driver's view while driving. The examples of the current fatigue monitoring systems are Electroencephalography (EEG) and Electrocardiography (ECG), which uses the frequency of brain and measures the rhythm of heart to detect fatigue level respectively. Currently, there are previous study of drowsiness detection system that uses IP camera or webcam, however the systems are not integrated with GPS and applied as IoT. Since there is no online fatigue detection tool, the current detection tools available are bulky and lack of interconnectivity.

This project aims to create online, responsive and real-time fatigue system that can detect and analyze multiple parameters of human signs of fatigue, integration process of multiple data from multiple sensors which might have different formats requires the data to be converted and synchronized. With few numbers of connected devices, it is required to possess complex or big data processing to extract meanings from the varied, enormous and continuously expanding data and information. The implementation of multi-sensor system may also contributes to the project complexity that need to be countered by the author. At the end, this project will help to enrich users experience and also help to enable new business processes, services or models.

1.3 Objective and Scope of Study

In this project, the IoT technology will be applied and tested to create fatigue monitoring and alert systems which incorporates sensors and internet to detect driver condition and location. Both of the data will be integrated to give alarm and suggest the closest safe and suitable area to rest and reenergize. Thus, this project will mainly focus on the connected devices in Internet-of-Things (IoT) study to create the alert systems. The objectives of the project will be:

i. To collect and process multiple data from sensors

ii. To formulate the driver fatigue criteria
iii. To integrate and process multiple data for driver condition and location
iv. To design an alert system to notify the driver of the current body fatigue condition

From all the objectives above, it is required for the student to acquire knowledge in Image Processing and Communication Systems. It is expected for the student to complete this project within eight-months ending in May 2017.

CHAPTER 2.0

LITERATURE REVIEW/THEORY

2.1 Drowsiness and Fatigue State

As stated by Gianluca Borghini *et al*, [1] mental fatigue is a drowsiness factor that may cause the person who experiences it to be unable to perform tasks. This is because the performance efficiency of human brain has decreased especially to respond to sudden events. While another source [2] mentioned that drowsiness is a state whereby an individual is in between of sleepy and awake condition, which also drawn the body and mental performance ability, Antoine Picot *et al*.

In Malaysia, a number of accidents occur because drivers tend to force driving though in drowsy or fatigue condition. The impact of the accidents are highly fatal if it involves heavy vehicles such as truck, lorry, bus, trailer and high speed car. Usually, an investigation will be conducted to know the factor that leads to the accident, for example alcohol, drug or other health-related tests. However, if an accident is caused by sleepiness or fatigue, police usually will have difficulties to identify if the two factors really are the significant cause to the accidents [3].

2.2 Drowsiness and Fatigue Detection by Human Face

This project will utilize human face to detect drowsiness and fatigue felt by drivers while on road. As stated in [4], [5] and [6], these two condition can be detected by extracting certain data from human face parameters. For an example, from [5], drowsiness can be easily detected by using eyelid movement of the driver. It is mentioned that eyelid has four types of movement which are completely open, completely closed, middle of open to close and middle of close to open. In order to extract the data, it is required to specify exact range of measurement for all the four eyelid movement types.

Besides eyelid, mouth can also be the parameter to detect fatigue or drowsiness [7], [8]. As mentioned in Scientific America by Associate Professor of Physiology and also the Independent Study Program Director of Osteopathic Medicine in Lake Erie College [1], having insufficient level of oxygen of the brain hypothalamus, specifically in the paraventricular nucleus (PVN) may cause yawning [4]. Having low oxygen level in the body is one of the signs that the person needs to rest.

In addition to ensure the detection is accurate, another parameter is preferably to be added in the system. Also, since eye detection is difficult for drivers wearing sunglasses, head nods or head positioning parameter can be used [9] since a drowsy driver will also experience frequent head nods.

2.3 Drowsiness and Fatigue Detection by Electroencephalography and Electrocardiography

Electroencephalography (EEG) uses imaging technique to analyze data from brain structures that generate scalp electrical signals. There are two measurement methods that can be done by EEG, *electrocortiogram* is the one which is directly measured from cortical surface while *electrogram* measures using depth probes. However this project will specifically emphasize only on EEG that uses the head surface measurement. This method is acknowledged as completely non-invasive technique if it is to be conducted many times to an individual with no virtual limitation and risk [10], [11] EEG can evaluate types of brain disorders including evaluating epilepsy [8], a state where brain gives sign of abnormal electrical activity or human body experience sensory disturbance, consciousness losses, or convulsions.

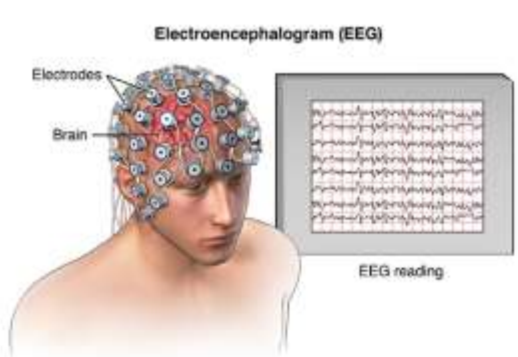


Figure 1 EEG applies special equipment and software to obtain and analyze the data from brain [11]

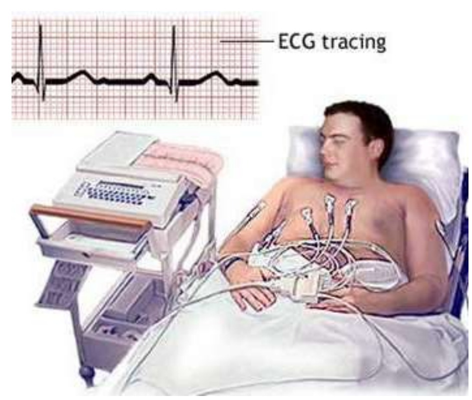


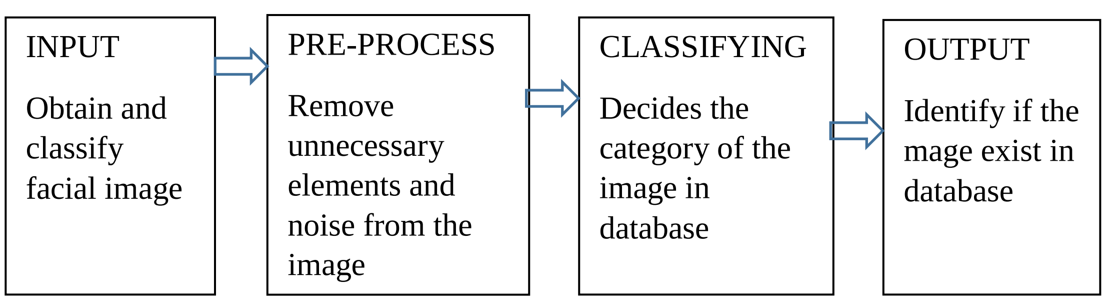
Figure 2 ECG analyze human body by using heart activities [10]

An electrocardiogram (ECG) analyzes electrical activity of heart to evaluate for many body disorders including sleep apnea where people stop and restart breathing few times when asleep and congestive heart failure that can be experienced from irregular heartbeat, fatigue, and shortness of breath [12]. Both of ECG and EEG can give very detailed results, however, both of these methods require long time to obtain result, expensive and complex equipment and expertise to conduct the test [11].

2.4 Image Processing and Viola-Jones Facial Identification Algorithm

Recently, digital image processing has been widely used to recognize facial features, for example in the application of automatic facial recognition in intelligence and surveillance system, catalogue of image database, and health status identification in medial field. This process involves two stages where initially it will identify the area that human skin presents in color image and then followed by detecting face and extracting data from the area. This facial feature detection is done through grey-scale image and the integration of mathematics morphology and thresh-holding [13].

In detecting facial features image, problems may arise due to complexity or variety of the images obtained resulting from the difference of skin structure and color, gender, head position. This problem may be broaden by different image quality, expression of face, and the lighting [14]. Thus, the images obtained must be pre-processed prior to extracting data from it as below:



To improve the result further, Viola-Jones Face Detection Algorithm may be implemented, which involves image integration to compute facial features, Adaboost to select feature and direct cascade to compute the source distribution efficiently [14].

2.5 Internet-of-Things (IoT)

The IoT (Internet-of-Things) is comprised of the combination of electronic devices, data, sensors and actuators, processes and network system to work. The example for IoT applications are smart home, smart city, and smart vehicle [15]. Among the advantages of IoT are, it can be used at any time, any devices, anywhere, any network, for any services or products. In this project, the IoT will be implemented in order to create a smart device for drowsy or fatigue drivers. The figure below shows the process involved in IoT.

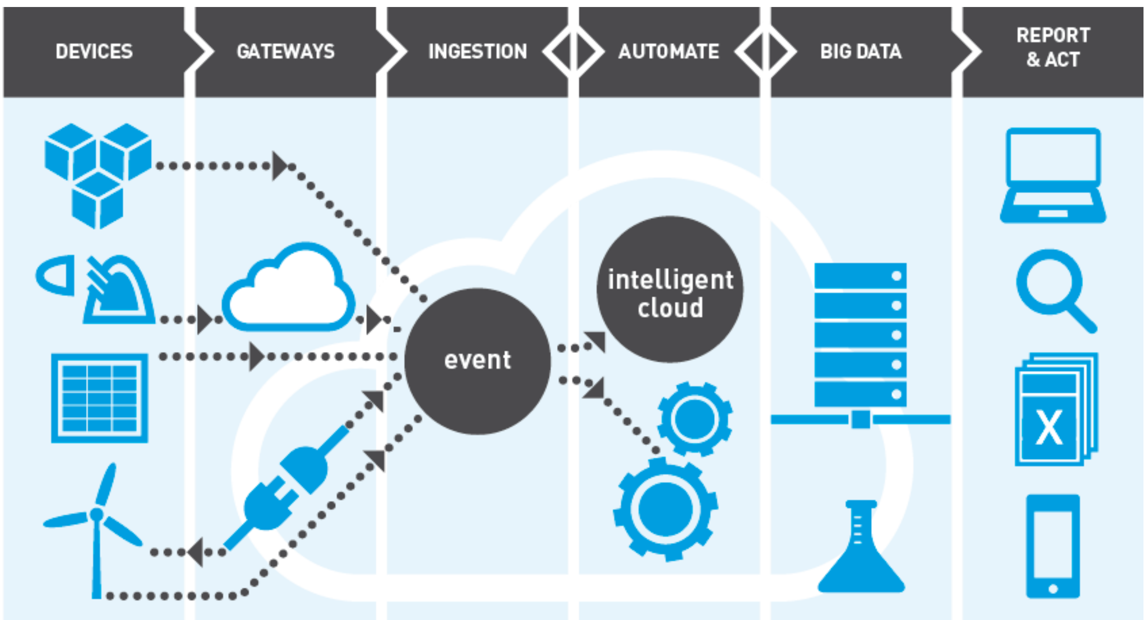


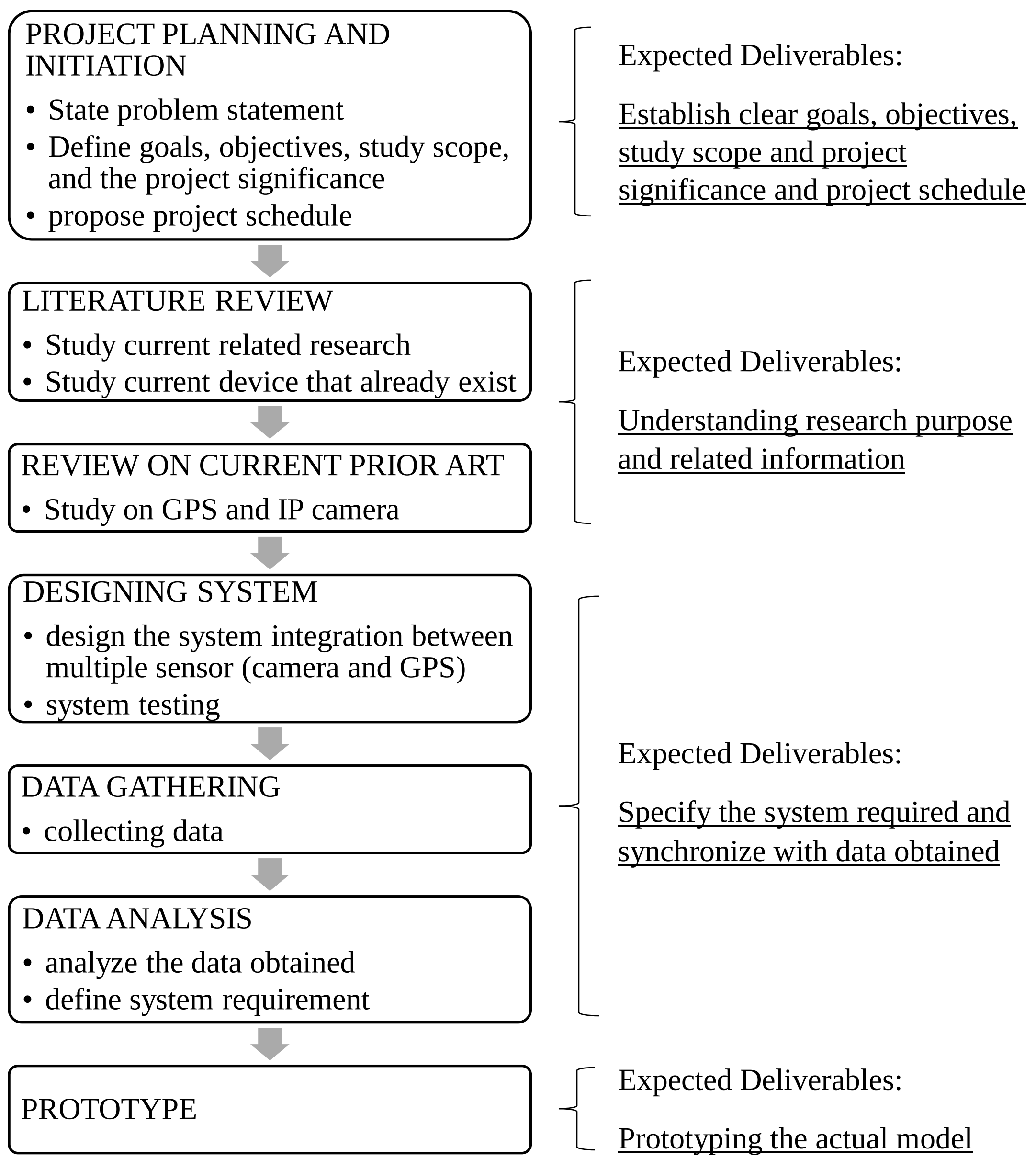
Figure 3 The IoT Value Chain [8]

CHAPTER 3.0

RESEARCH METHODOLOGY/PROJECT WORK

3.1 Proposed Research Methodology, Flow Chart and Expected Deliverables

To ensure this project is organized in a systematic way and manage to achieve its goals within specified time range, the procedure and activities required for execution have been planned and put into flowchart as below:



For the first stage which is Project Planning and Initiation, clear objectives have been established as stated in section 1.3. It focuses mainly on the connected devices in Internet-of-Things (IoT) study to create the alert systems. Initially, it is required to collect and process multiple data from sensors and formulate the driver fatigue criteria to enable the integration and process multiple data for driver condition and location. Then, an alert system will be designed to notify the driver of the current body fatigue condition. In Literature Review, it is required to perform a critical on the analysis done for each literature review. This is in order to obtain most current, temporary, precise and relative information about the past researches that have been conducted related to the system. For the Review on the Current Prior Art stage, it is assumed that the performance and efficiency of the devices used such as Webcam and IP Camera has almost equal resolution but may have different function.

While Designing the System, it is important to note that consistent parameters and device specifications are used. For Data Gathering process, each data must be collected in the same surrounding and condition for an instance, on day or night time, and on healthy person as person with illness will tend to be fatigue all time which will cause the data to be invalid. Then, only valid data is used to be analyzed and last but not least, to create the prototype by integrating each subsystems which involves sensors, networks, devices and programming codes.

3.2 Project Gantt-chart and Key Milestone

Below is the scheduled Gantt chart for this project in semester 1. Included with the planned activities are the key milestone for the whole semester:

ACTIVITIES	WEEK													
	1	2	3	4	5	6	7	8	9	10	11	12	13	**14**
SEM 1														
Project Identification														
Selection of Project Title	■													
Identifying Objectives and Scope of Study		■												
Preparing Project Ganttchart, Key Milestone, Process Flow			■											
Discussion on Fundamental Knowledge				■	■	■								
Interim Report Preparation														
Literature Review			■	■	■	■								
Collecting Data Material				■	■									
Purchase Material					■									
Formulate Driver Fatigue Criteria **[O2]**					■									
Drafting Extended Proposal			■	■	■									
Submission of Extended Proposal						★								
Learn Image Processing and Programming							■	■	■	■	■	■		
Proposal Defence and Progress Evaluation									★					
Drafting Interim Report										■	■	■		
Submit Interim Report (to SV)													★	
Submit Interim Report (to Examiner)														★

Figure 4 Project Gantt-chart and Key Milestones for Phase I

While below is the scheduled Gantt chart for this project in semester 2. Included with the planned activities are the key milestone for the whole semester:

ACTIVITIES	WEEK														
	1	2	3	4	5	6	7	8	9	10	11	12	13	**14**	**15**
	SEM 2														
Collect and Process Data from Different Sensors **[O1]**	■	■													
Integrate and Process Multiple Data for Driver Condition and Location **[O3]**		■	■												
Design an Alert System to Help Driver **[O4]**				■											
Analysing Data					■										
Progress Report Submission for reviewing by SV						■									
Submit Progress Report							★								
Poster Preparation							■	■							
Pre-Sedex										★					
Submit Draft Final Report											★				
Technical Paper Writing										■					
Submit Technical Paper for reviewing by SV											■				
Submit Technical Paper to Examiner												★			
Dissertation Writing															
Dissertation submission (softbound)												★			
Viva													★		
Submit (hardbound)															★

Figure 5 Project Gantt-chart and Key Milestones for Phase II

3.2.1 Implementation Plan (FYP II)

WEEK	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14
ACTIONS													
Identify parts that need help (coding, gps integration)	■												
Find people/sources that can help to solve (Dr.Norashikin/Dr.Nasreen etc)		■											
Process the data (image) by using matlab (fatigue parameters identification: eye, mouth)		■	■										
Process location of UTP and RnR in Perak with the database from internet (obtain latitude, longitude and shortest distance)		■	■										
Testing and integrating the GPS chip with computer	*waiting for item to arrive (test with google map data while waiting for GPS chip)*												
Process GPS data and integrate with Matlab (temporarily use data from google)				■	■								
Interfacing the IP camera and GPS chip together (with Matlab)	*waiting for item to arrive (test with google map data while waiting for GPS chip)*												
Calibrating through programming to get accurate result of fatigue/drowsiness level				■	■	■							
Display the output						■							
Touch up											■		

Figure 6 Implementation Plan for FYP II

3.2.2 Project Key Milestones (FYP II)

NO	ACTIVITIES	WEEK	STATUS
1	Project Topic Selection : Connected Devices in Internet-of-things (IoT) for Driver Fatigue Alert System (16S041)	1	DONE
2	Identifying and purchasing the suitable components and software for the project	3-5	DONE
3	Formulate driver fatigue criteria	5	DONE
4	Testing and integrating the IP camera with computer	7	DONE
5	Identify and test the best method to process the image data by using Matlab	12	FYP II
6	Testing and integrating the GPS with computer	16	FYP II
7	Process GPS data and integrate with Matlab	18	FYP II
8	Interfacing the IP camera and GPS together (with Matlab)	20	FYP II
9	Calibrating through programming to get accurate result of fatigue/drowsiness level	23	FYP II
10	Display the output	25	FYP II

Table 1 Project Key Milestones for FYP II

As for Key Milestones, in step 2 where suitable components and software are chosen to be applied after some survey and studies are done to look for its efficiency and its importance. For an example, Video Converter is important to ensure the video is able to be processed by Matlab and IP Camera is used instead of Webcam because it is portable, and easier to be connected to networks and other devices even though the IP address is changing. While formulating driver fatigue criteria, it is assumed that every person have same signs of fatigue which involves eyes closure, yawning and head tilting.

CHAPTER 4.0

RESULTS AND DISCUSSIONS

4.1 Tools

The tools which includes the hardware and software to be used in this project and the reasons of the usage are as listed below:

Hardware		Software	
Type	Judgement	Type	Judgement
1. Router (for networking)	Both computer and camera need network to receive and transfer data	Matlab (Image Processing)	Has many programming references
2. Go-q IP Camera	▪ Detailed manual user instruction ▪ Can install phone apps, easy to be used as portable device ▪ Have criteria that are compatible with Matlab ▪ Has flexible turnaround degree	Video/Image converter	If the image is not in mpeg format
3. GPS Chip	▪ Has USB interface to program with Matlab	Video cutter	If required
4. Interconnected Devices (Laptop & Cellphone)	▪ Portable ▪ User-friendly ▪ Easy to connect with any networking systems	Mapping System (google map database)	The database is an open source

Table 2 List of Hardware and Software Used in the Project

Figure 7 Go-Q IP Camera

Type:	WiFi IP Camera 720P
Model No./Model:	QF002
Resolution:	1280*720P
Sensor:	1/4", CMOS
Video Compression:	H.264
Day & Night:	Auto
Min Illumination:	Olux (IR ON)
WiFi Protocol:	WiFi 802.11 b/g/n
Motion Detection:	Support
ONVIF:	Support

Table 3 Specification of Go-Q IP Camera

4.2 Systems Parameters and Flow Chart

The systems parameters include the driver fatigue criteria by facial parameters as shown below. However, the author will only focus on 2 parameters which are eyes closure percentage and the mouth opening. The head position will only be implemented based on time availability.

PARAMETERS	DESCRIPTION

Eyes closure percentage	- By using PERCLOS algorithm → measure the proportion of total time that eyelids are closed 80% or more - If eyes were closed >310.3ms (normal eye blink)
Mouth opening (yawning)	- If mouth was open>6s (normal yawn)
Head position	- If head is tilting 45° to right side/-45° to left side/45° to front more than 3s

Table 4 Parameters Used in the Detection System

The system process flowchart will be as below and might be changed according to future experiment observations or problems that might result:

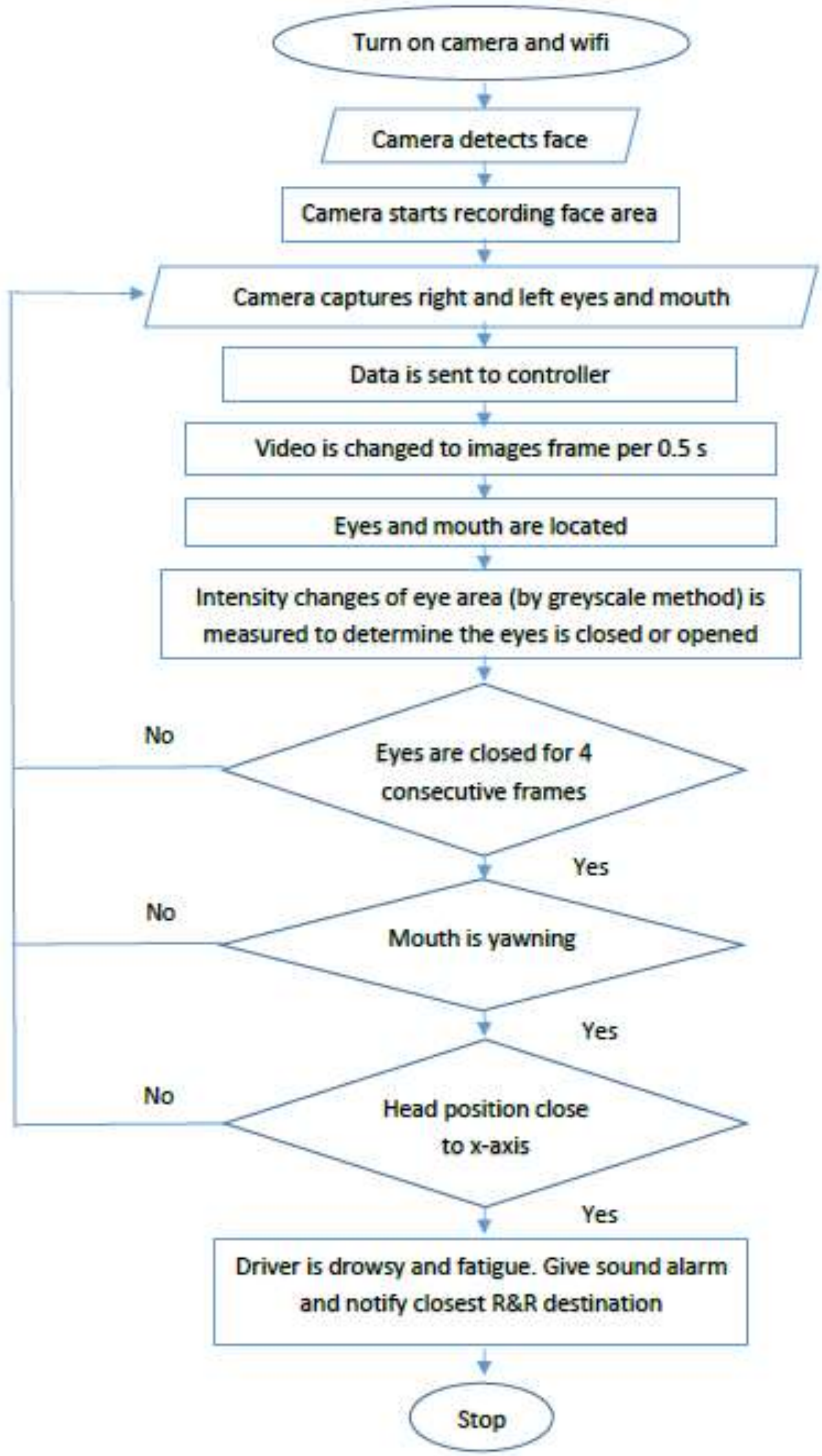
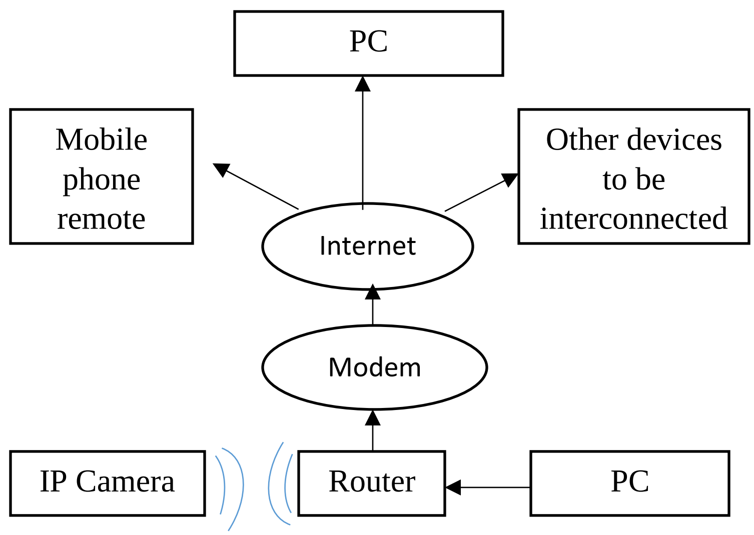


Figure 8 System Process Flowchart

4.3 Integration of Go-Q Ip Camera with Network Connection and Devices

The connection of the IP Camera with the network and other devices such as computer and phones are as below, where the network can be either from WIFI or LAN. This camera will use wireless network to connect to interconnecting devices and also to send data.



Below is the network specifications for the camera:

- Protocol: DDNS,DHCP,DNS,FTP,HTTP,NTP,P2P,RTSP,SMTP,TCP,UDP,UPNP
- Wireless: WiFi 802.11 b/g/n
- IP Mode : Dynamic IP address, static IP address
- Online Visitor (Max.): 4
- Safety: Administrator password protection

General specifications for the camera:

- Rated power: 3.5W (with IR opening)
- Maximum power: 7W (P/T working)
- Specification of Power Supply: DC 5V (± 0.3V)
- IR CUT dual filter
- Infrared Distance: 10m
- Mobile Access: Android, iPhone OS
- H.264 high profile

Video / Image / Audio

- Video Compression Format: H.264
- Video Standard: NTSC,PAL
- Frame Rate (FPS): 30fps
- Resolution: 1280 x 720
- Infrared Sensitivity: Yes

Figure 2 below shows how to configure the router and set up the wireless network so that all the devices can be interconnected with each other to transmit and receive data. While Figure 3 shows how to configure the IP camera and view the image on desktop. While Figure 4 shows the clear image that is taken by the IP camera and is directly sent to the devices connected to it, either cell phone, or laptop.

Figure 9 Configuring the Router

4.3.1 Connecting the IP Camera and Other Devices with Network

i. Discovering the IP Camera

After the IP address of the IP Camera is discovered, the author tried to ping it from the computer to test its connection with the network and either the computer can reach the IP Camera. Then, the computer and the IP camera must have same first 3 segments (octets) numbers which means that they are connected on the same network. As if the computer manages to receive feedback from the camera, this means that both are successfully connected to each other on the same network.

Thus, both will able to be connected via web interface by using their respective IP address.

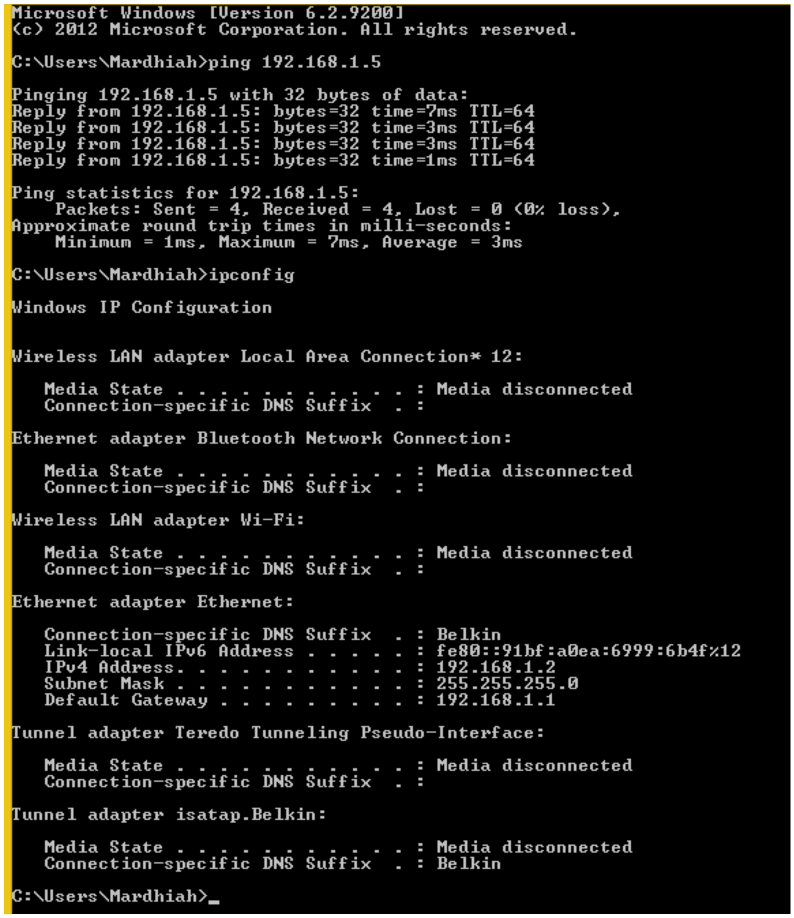


Figure 10 Pinging the IP Camera

Figure 11 Clear Image Viewed through Device in .jpg format

Figure 12 Configuring the IP address of the IP Camera and Viewing the Image

4.3.2 Integrating the IP Camera with Matlab

The IP Camera can be connected by declaring the *ipcam*. The *url* of the camera is needed to create the object. First, the camera is connected without user authentication. The *url* must be *.mjpeg* (Motion JPEG) HTTP or RTSP stream, so the URL must start with http. After the object is created, the image and video can be previewed and snapshots can be obtained from the camera. Below is the script from matlab with the IP camera *url*.

```
cam = ipcam('http://192.168.1.5/video.mjpeg')

url = 'http://192.168.1.5:80/jpg/image.jpg?timestamp=';
image  = imread(url);
fh = image(ss);
while(1)
  image  = imread(url);
  set(fh,'CData',image);
  drawnow;
end
```

Then, the stream from the IP Camera can be extracted frame by frame, however by using this, a problem that may be encountered is the slow streaming rate.

4.4 Algorithm Design and Development to Detect Fatigue Parameters

As the means to detect the parameters of eyes closure, mouth opening and head position, certain algorithm and method have been applied and it is named as Cascade

Object Detector. This method uses Viola-Jones algorithm in order to detect and define the measurement of eyes closure, mouth opening and head position.

4.4.1 Face Detection Algorithm (Viola-Jones)

The framework of Viola-Jones object detection especially image, is used to identify many classes of objects. However, this algorithm is more applicable for facial features detection. It uses the rectangle features concepts that measures the pixel sum in rectangular region. To elaborate more, the pixel sums which exists in white rectangle area is deducted from the pixel sum in grey-colored rectangle. The *two-rectangular feature* value represented by A and B is the sum or difference of the pixel amount in the two rectangular sections. These sections will have equal shape and size dimension, and also are either vertically or horizontally oriented while being adjacent with each other.

While the *three-rectangular feature* labelled as C is the value or summation computed in two outer rectangles deducted from the sum in the central part of the rectangle. The *four-rectangular feature*, labelled as D calculates the rectangles diagonal pairs difference.

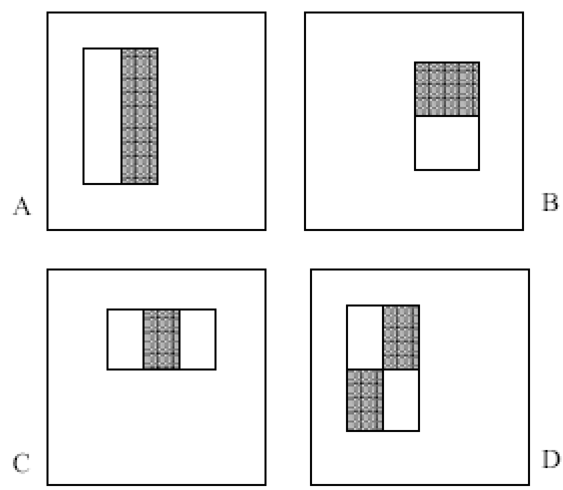


Figure 13 Four features in Viola-Jones Algorithm

These rectangular features can be rapidly calculated by using image representation called image integral.

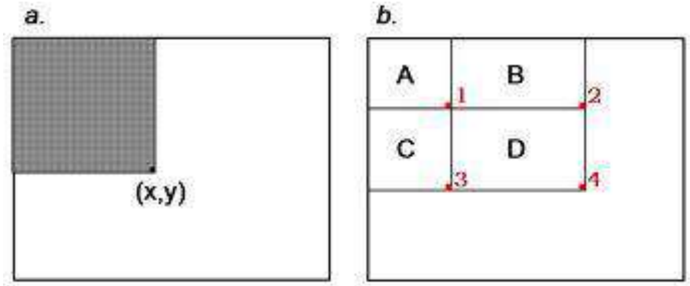


Figure 14 Image Integral

The integral image value at point (x, y) is the pixels sum from above to the left side. Based on the value obtained, the pixels sum in section D is calculated by using 4 references of array. The integral image value at point ‘1’ is the pixels sum in section A. While the values at point ‘2’ is equal to A + B, at point ‘3’ is equal to A + C, and at point ‘4’ is equal to A + B + C + D.

To detect face, the methods used to detect certain facial parameters are shown in Figure 10. There are two features extracted from face; top row (overlay with bottom row). The first feature detects and computes the intensity difference between the eyes section and upper-cheeks section. It is due to the eyes section is typically, darker than upper-cheeks and nose. While the second feature will detect and compare the difference between intensities of eye sections and across the nose bridge.

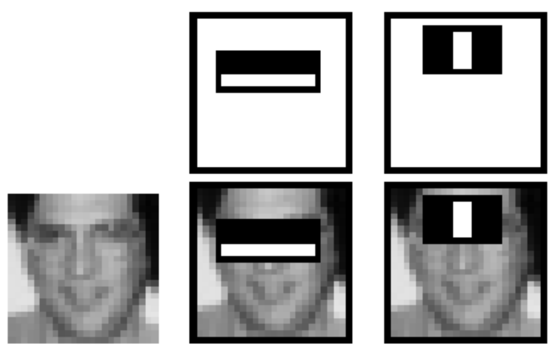


Figure 15 Extraction of Face Detection Features

4.4.2 Classifiers Cascade

It is assumed that there are around 45,396 possible features that can be detected in a 24x24 pixel standard sub-window. As it seems a very big number, it might prohibitively difficult or complex to be processed and evaluated. Thus, to improve the detection process and performance, the features measured can be added to cascade of classifiers. Though, this method may directly cause the computation time to increase and result in slower detection process. The classifiers cascade is customized to improve detection process and performance and at the same time, to reduce the total computation period.

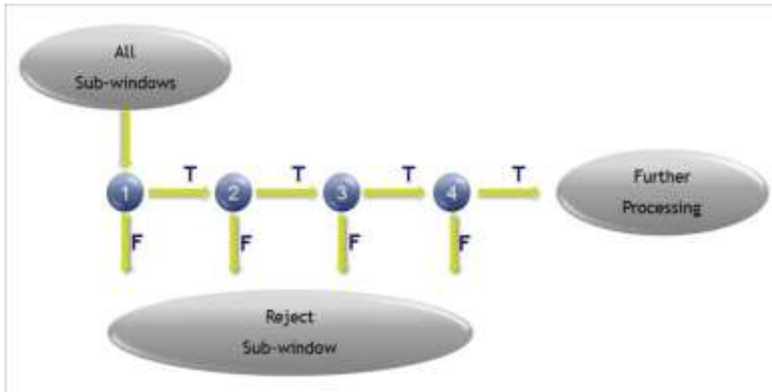


Figure 16 Cascade of Classifiers

The implementation of this method can be performed quickly, however this method might not be able to perform very fast for real time application. This is the reason why strong classifiers are constructed in a way of cascade system, to reduce its complexity. Every successive classifier is set to process only for selected samples or data, which then passing across the preceding classifiers. In certain cases, the classifier (in any of the stages) rejects the sub-window that is under inspection, further processing will not be performed and it will proceed to look for next sub-window.

4.5 Developing Source-code

4.5.1 Facial Detection

The facial detection is done by Vision Cascade Detector which is part of Computer Vision Toolbox System which detects object by using Viola-Jones method. The figure below shows the script run in Matlab, however due to programming errors, the result cannot be obtained. However, the result of the detection features should be obtained as below:

1. The cascade object detector is defined and set up by constructor. Then, the constructor uses built-in Viola-Jones algorithm to recognize and detect the facial features such as eyes, mouth and head position.

2. The videos (or the image) selected are read and then the face detector is started to run. The data (video or image) are directly sourced from the IP camera.

3. The bounding box (rectangle) appears around the face detected which covers the area of the facial parameters.

Editor - C:\Users\Mardhiah\Desktop\codes\detectingface.m*

detectingface.m*

```
% create cascade detector object
FaceDetector = vision.CascadeObjectDetector();

% read video frame and run the detector
VideoFileReader = vision.VideoFileReader('Video1.mp4');
VideoFrame      = step(VideoFileReader);
bbox            = step(FaceDetector, VideoFrame);

% draw the returned bounding box around the detected face
VideoOut = insertObjectAnnotation(VideoFrame, 'rectangle', bbox, 'Face');
figure, imshow(VideoOut), title('Detected Face');
```

Figure 17 Command Script for Facial Detection

Figure 18 Face Detection with Bounding Box (bbox)

4.5.2 Eyes Detection

The eyes detection is also done by Vision Cascade Detector and the same algorithm of Viola-Jones method, however in this step, the object is the eyes. The eyes are suggested to be separately detected to get more accurate result especially when the driver is tilting his head. A problem may result where other facial parts are detected as eyes, which is considered as False Positive result while True Positive is when the actual eye area is detected within the box. The False Positive result occurs when condition is normal but wrong eye area is detected. However, the author cannot present the result due to programming errors. The result of the detection features should be obtained as below:

1. The cascade object detector is defined and set up by constructor. Then, the constructor uses built-in Viola-Jones algorithm to recognize and detect eyes

2. The videos (or the image) selected are read and then the detector is started to run.

3. The bounding box (square) appears around the face detected which covers the area of the facial parameters (left and right eyes)

4. True Positive results when correct eyes are detected in normal condition and False Positive is when the area other than eyes are detected

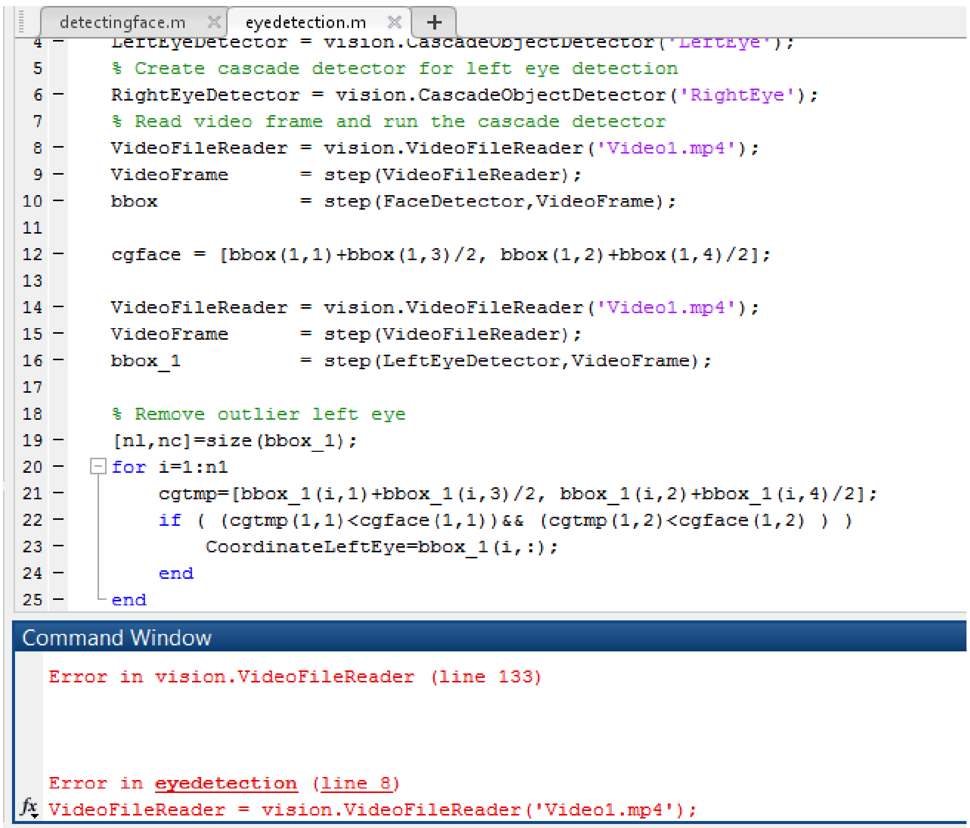


Figure 19 Command Script for Eyes Detection and the Errors Detected

Figure 20 Expected Result of Eyes Detection [20]

4.5.3 Lips (or Mouth) Detection

In mouth detection, the same Viola-Jones algorithm has been applied in cascade object detector to detect the object (lips in this step) in a rectangle box. The algorithm is somehow expected as not very efficient as how it locates the other parameters. The result cannot be viewed due to programming errors. However, the features should be as below:

1. The constructor will define and setup the cascade object detector. It will apply Viola-Jones built-in algorithm to detect the lips boundary.
2. The video (or image) which is selected is read and the detector is run.
3. A bounding box is located to surround the lips. The bounding box (bbox) indicates the desired lips area.
4. There will be multiple bbox appears as it detects multiple mouth (actual and artificial).
5. The mouth detected will then be classified as either True Positive or False Positive.

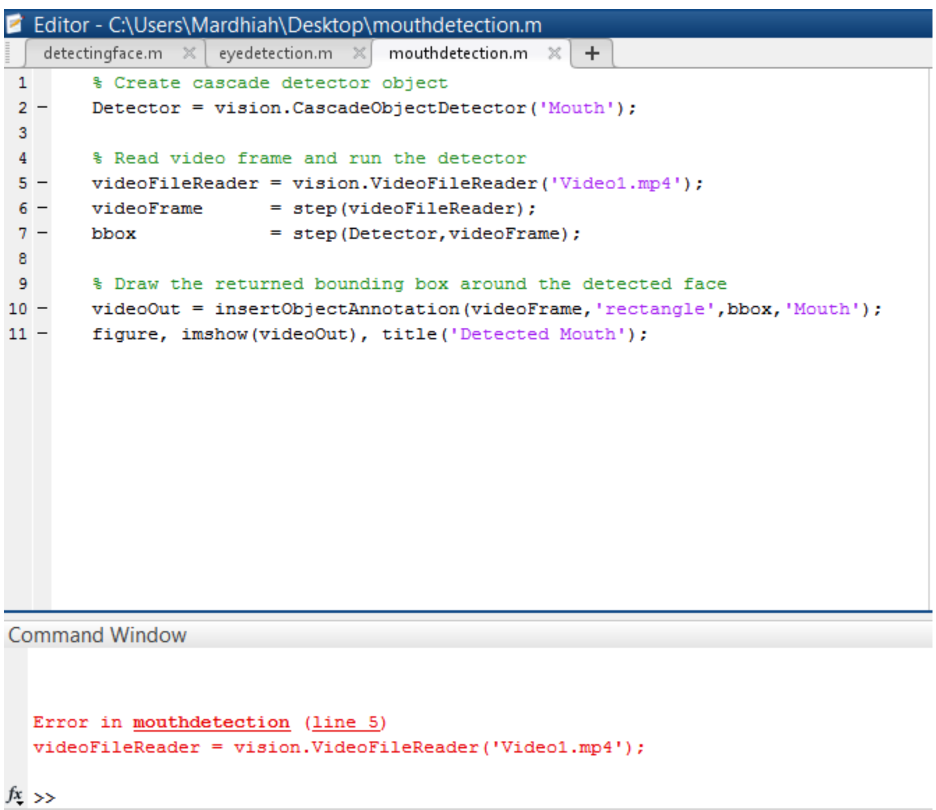


Figure 21 Command Script for Lips Detection and the Errors Detected

Figure 22 Expected Result of Lips Opening Detection [21]

4.5.4 Test Analysis

In order to ensure the accuracy and flexibility of the drivers condition test result, it is important to install the camera in its most suitable location so that it manages to detect the parameters (face, mouth and lips) although the drivers is not in static position. Shown below is the result after few tests have been conducted. The result accuracy has been compared to the actual result taken from manual snapshots.

Detection	Face	Mouth	Yawning
Camera under mirror	85%	50%	22%
Camera on dashboard	93%	78%	60%

Table 5 Accuracy According to Camera Installation

From the image data, drowsiness level indication analysis should be obtained as shown in the figures below:

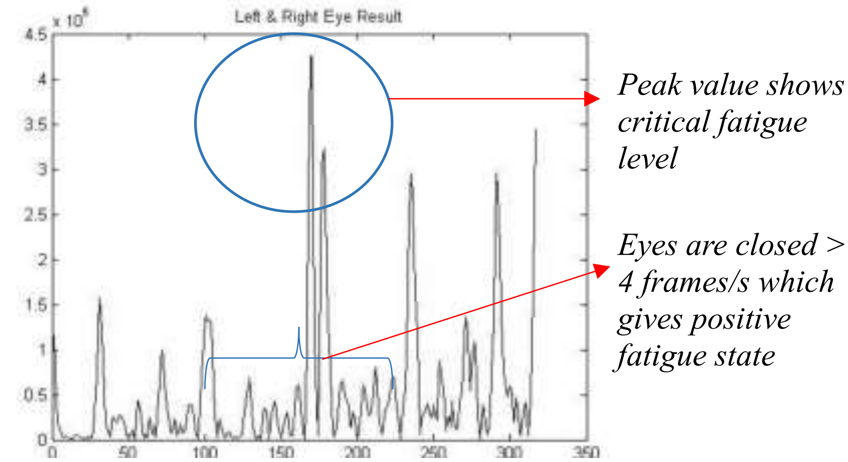


Figure 23 Drowsiness Indication for Eyes [22]

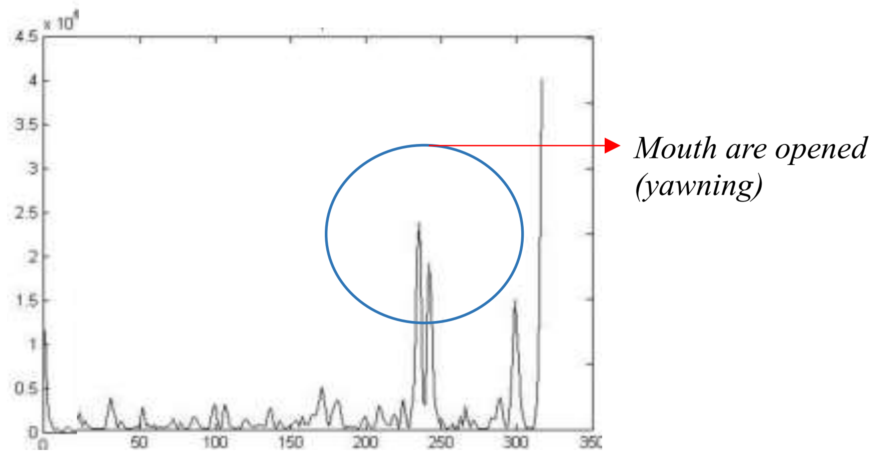


Figure 24 Drowsiness Indication for Mouth [22]

As can be seen in Figure 18, there are two indications that can be analyzed for fatigue level, which are the peak value and four consecutive frames per second which shows positive fatigue state of the driver. This indicates that the proportion of total time that eyelids are closed 80% or more is more than 310.3ms (normal eye blink) or 4 frames/s. Whereas for Figure 19, the circle shows the initial sign of yawning which is more than 2s while normal yawn is 6s in normal condition.

4.5.5 Obtaining Location Data from Google Map

The author applies Google Maps API to mapping the current figure with Google Maps database. Technically, it is assumed that the coordinates (latitude and longitude) for current location (figure) are as in the specified WGS84 datum. It applies certain code of conversion for converting and projecting the data from the system coordinate used by Google into WGS84 coordinates. The short code below will result the image screenshot:

```
lat = [48.8708 51.5188 41.9260 40.4312 52.523 37.982];
lon = [2.4131 -0.1300 12.4951 -3.6788 13.415 23.715];
plot(lon,lat,'.r','MarkerSize',20)
plot_google_map
```

```
detectingface.m  eyedetection.m  mouthdetection.m  ge_region.m*

function kmlStr = ge_region(north,south,east,west,varargin)

AuthorizedOptions = authoptions( mfilename );
id              = 'region';
minAltitude     = 0;
maxAltitude     = 0;
minLodPixels    = 0;
maxLodPixels    = -1;
minFadeExtent   = 0;
maxFadeExtent   = 0;

parsepairs %script that parses Parameter/Value pairs.
kmlStr = [ '<Region id="' id '">' 13 ...
           '<LatLonAltBox> ' 13 ...
           '<north>' num2str(north,'%.6f') '</north>' 13 ...
           '<south>' num2str(south,'%.6f') '</south>' 13 ...
           '<east>' num2str(east,'%.6f') '</east>' 13 ...
           '<west>' num2str(west,'%.6f') '</west>' 13 ...
           '<minAltitude>' num2str(minAltitude) '</minAltitude>' 13 ...
           '<maxAltitude>' num2str(maxAltitude) '</maxAltitude>' 13 ...
           '</LatLonAltBox>' 13 ...
           '<Lod>' 13 ...
           '<minLodPixels>' int2str( minLodPixels ) '</minLodPixels>' 13 ...
           '<maxLodPixels>' int2str( maxLodPixels ) '</maxLodPixels>' 13 ...
           '<minFadeExtent>' int2str(minFadeExtent ) '</minFadeExtent>' 13 ...
           '<maxFadeExtent>' int2str(maxFadeExtent ) '</maxFadeExtent>' 13 ...
           '</Lod>' 13 ...
           '</Region>' 13 ];
```

Figure 25 Script Command for Defining Region for Driver Destination (North, South, East and West) through Google Earth

4.5.5.1 Signal Acquisition from Satellite

Generally, to receive signal of GPS, the process involves three steps which are: Acquisition, tracking, decoding data and position solution. The explanation of the steps are as discussed below but for this part, only signal acquisition and tracking are discussed:

1. Acquisition

This signal process involves 3-dimensional searching in 3 parameters: time (code phase), frequency and satellite-specific PRN code.

There is 2 methods of signal acquisition: serial search and FFT search

2. Tracking

Then, receiver goes to 2nd mode which is tracking. Delayed Locked Loop (DLL) is applied for frequency tracking. While Phase or Frequency Locked Loop (I'LL/FLL) are applied for carrier phase/frequency tracking. MATLAB applies DLL for loop tracking.

3. Data decoding and position solution

After bothof the loops for tracking complete, 50 bps data will be decoded to determine position of receiver.

While the code for data acquisition is as below:

```
function [yes, f, ph] = acquisition(fid , K, sat_id, fc_lo, step, fc_hi, T)
 % The function
% [yes, f, ph] = acquisition(fid , K, sat_id, fc_lo, step, fc_hi, T)
% perform an aquisition process for one satellite.
% RETURNED :
 % yes = 1 if the satellite signal is acquired
% yes = 0 if the satellite signal is NOT acquired
% f = carrier frequency
% ph = code phase (samples)
% ARGUMENTS :
% fid = file pointer to data
% K = number of sequential correlations to be accumulated
% sat_id = satellite to search for
%fc to = IF freq minus max Doppler offset
%fc hi = IF freq plus max Doppler offset
% step = size of one Doppler bin
% T = threshold (typically 6 if K = 10)

%Initialize variables
i=0;
C = zeros(length(fc_lo:step:fc_hi), 5000);
% Doppler loop
for fc = fc_lo:step:fc_hi
 i = i+1;
 % K sequential correlations is accumulated
 for j=1:K
```

```
   % Read next 5000 samples
   data=(fread(fid,5000,1schar1))1;
   C(i,:)=C(i,:)+fftsearch(data, sat_id, 1.023e6, 5e6, fc, 5000);
  end
  % Go back to beginning of file
  fseek(fid,0,-1);
 end

 % Check threshold
 yes=max(max(C))/mean(mean(C))>T;

 if yes % The satellite signal is assumed to be acquired
  fc_vector = fc_lo:step:fc_hi;
  % Find the maximum correlation
  n = find(C==max(max(C)));
  % Return the code phase
  ph = floor(n/length(fc_vector));
  % Return the carrier frequency
  n = mod(n,length(fc_vector))
  if -n
   n = 1;
  end
  f = fc_vector(n);
 else  % The satellite signal is not acquired
  % return zeros
  f = 0;
  code_phase = 0;
 end
```

Below here is the Matlab programming code for data tracking:

```
function Ip = track(fid, fc, ph, num_sec, sat_id)
% The function Ip = track(fid, fc, ph, num_sec, sat_id)
% returns the data output from the prompt I-channel
% of the code & carrier tracker.
% fid = file pointer to data
% fc = carrier IF
% ph = codephase (samples)
% num_sec = how many seconds of data to process
% sat_id = what satellite to receive

% SETUP
% # of samples between early-prompt and late-prompt channel
chip_delay=3;
% code chip rate
fcode = 1.023e6;
% # of samples in a code period
scode = 5000;
% # of codeperiods to average over
num_ava = 20;
% Set file pointer to code phase
fseek(fid,Oh,-1);
% Initialize variables
y(1)=0;
pnco=0.0;
t_end = 0;

```

```
% Filter coefficients
81=50;
damp=.7;
wn=2*B1/(damp+1/(4*damp));
dT=le-3;
K=400*pi;
ci = 1/K*8*damp*wn*dT/(4+4*damp*wn*dT*(wn*dT).-2);
c2 = 1/K*4*(wn*dT).-2/(4+4*damp*wn*dT*(wniodT).-2);

% generate CA codes
prompt=cacode(sat_id,fcode, 5e6, scode);
early=[prompt(scode+i-chip_delay:scode) prompt(1:scode-chip_delay)];
late=[prompt(chip_delay+1:scode) prompt(1:dhip_delay)];

% Loop
for i=1:round(num_sec*1000/num_ava)
 % Averaging loop for code tracking
 for j=1:num_ava
   % recalulate time vector
   t = t_end:(1/5e6):(scode-1)/5e6+t_end;
   t_end = t(scode)+1/5e6;
   % generate I/Q (pnco = phase offset)
   Icomp_lo=sin(fc*2*pi•t+pnco);
   Qcomp_lo=cos(fc*2*pi•t+pnco);
   % read data
   data=(fread(fid,scode,ischar1))1;
```

```
    % Remove carrier
      Icomp=data * Icomp_lo;
      Qcomp=data * Qcomp_lo;
      % Calculate index for vectors
      inda = (i-1)*num_ava+j;
      % Despread with CA code
        % In-phase
      Ie(inda)=sum(early .* Icomp); % early
      Ip(inda)=sum(prompt .* Icamp); % prompt
      Il(inda)=sum(late .* Icomp); % late
        % Quadrature
      Qe(inda)=sum(early .* Qcomp); % early
      QP(inda)=sum(prompt .* Qcomp); % prompt
      Ql(inda)=sum(late .* Qcomp); % late
      % Envelops
      ee(inda) = (Ie(inda).-2 + Qe(inda).-2).-.5; % early
      PP(inda) = (Ip(inda).-2 + Qp(inda).-2).-.5; % prompt
      ll(inda) =            +                   % late
      % Costas loop for carrier tracking
        % Atan discriminator
      d2(inda)=atan(Qp(inda)./Ip(inda));
        % Loop filter
      if inda<2
        y(inda)=0;
      else
        % Filtered discriminator output
        y(inda)=y(inda-1)+(cl+c2)*d2(inda)-cl*d2(inda-1);
        % Phase for Carrire NCO
        pnco=rem(K*y(inda)+pnco, 2*pi);
      end
    end
    % Code loop discriminator
    E_L = sum(ee((indatnun_avatthinda)./11((inda-num_avatthinda))/num,ava;
    % Check for thresholds and phase shift
    % CA code if either threshold is crossed
    if E_L < .8
      prompt=fprompt(2:scode) prompt(1)];
      early=(early(2:scode) early(1)];
      late=[late(2:scode) late(1)];
    elseif E_L > 1.5
      prompt=fprompt(scode) prompt(1:scode-1)];
      early=[early(scode) early(1:scode-1)];
      late=[late(scode) late(1:scode-1)];
    end
end
```

As for the codes above, the result obtained should be as below:

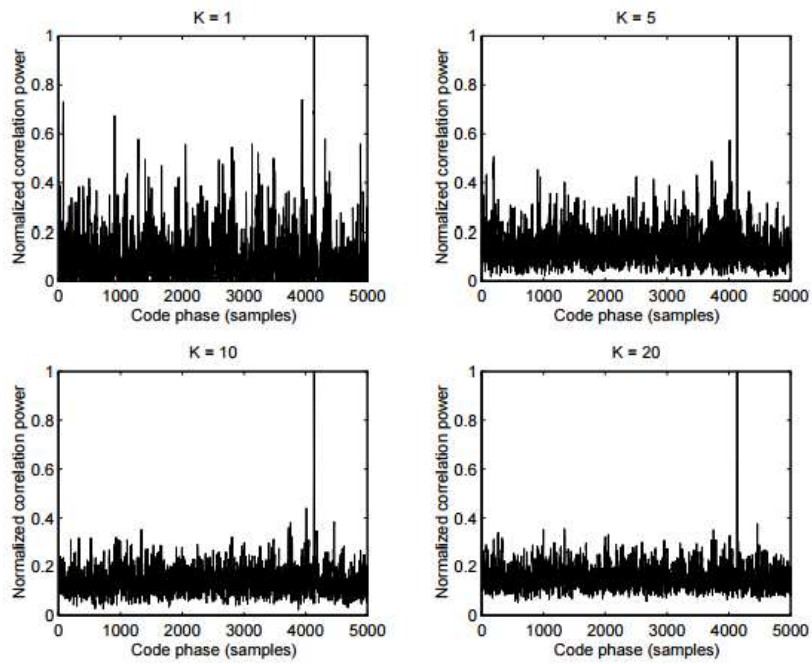


Figure 26 Signal Acquisition at 1.2445 MHz [23]

4.6 Overall Comment

The author is having difficulty to present the results mainly due to programming errors. Thus, an improvement need to be performed on the programming part in order to obtain the desired result and data before proceeding to the prototype. It is also important for the author to check on the file extensions used in Matlab. The codes are later to be integrated as a whole to create the system.

CHAPTER 5.0

CONCLUSION AND RECOMMENDATION

As for conclusion, this project may help drivers to prevent accidents due to fatigue and drowsiness experienced while on road. In Malaysia, accidents often occur due to two major factors, the negative driving habits or practice of the drivers and also the fatigue and drowsiness experienced. Some of the drivers do not even notice that their body need to rest and keep forcing to continue driving. Thus, it is good to create a system that can detect and notify the condition of the drivers. Whereas some drivers do notice they are fatigue or drowsy, they cannot estimate how far is the closest, safe and suitable location to stop and rest. This is why it is recommended to have the GPS and internet installed together in the system, to help the drivers navigate to the closest location.

In addition, as this project integrates network, data and interconnected devices or can also be referred as Internet-of-Things (IoT), it makes the prototype as a very user-friendly portable device that can be used by anyone at any time, with any devices especially mobile phones. People will have wide access to the system and it will be easier for people to integrate and rescale this system to other devices based on the expanding necessity.

For future project expansion, it is recommended to also design a system that can analyze the weather and traffic data and notify drivers about the situation. With the data being computed and analyzed, the result of how long will the driver arrive at the rest location will be more accurate. In addition to that, it is preferable to create this system in form of phone application which is more user-friendly and do not require any physical installation. It is also important to note that the device must be installed in suitable angle in front of the driver in order to not block the view that may also lead to accident. To also consider that the driver will travel at night, it is better to use infrared (IR) camera to obtain and analyze the image. Most of the previous accident tragedies occur at night where most people will choose to travel for long distance and visual view is not as good as day time.

References

[1]	G. Borghini, L. Astolfi, G. Vecchiato, D. Mattia and F. Babiloni, "Measuring neurophysiological signals in aircraft pilots and car drivers for the assessment of mental workload, fatigue and drowsiness," *Neuroscience & Biobehavioral Reviews,* 2012.
[2]	A. Picot, S. Charbonnier and A. Caplier, "On-line automatic detection of driver drowsiness using a single electroencephalographic channel," *Engineering in Medicine and Biology Society (30th Annual International Conference of the IEEE),* pp. 3864-3867, 2008.
[3]	G. Schultz. and H. Young, "A Safety Analysis of Fatigue and Drowsy Driving in the State of Utah," no. ES-1, 2007.
[4]	D. F. Dinges and ,. R. Grace, "PERCLOS: A valid psychophysiological measure of alertness as assessed by psychomotor vigilance," *Office of motor carriers, Tech. Rep. Federal Highway Administration.* , 1998.
[5]	D. Liu, P. Sun, Y. Xiao and Y. Yin, "Drowsiness Detection Based on Eyelid Movement," *Education, Technology and Computer Science (ETCS), 2010 Second International Workshop,* pp. 49-52, 2010.
[6]	J. Nagi, S. K. Ahmed and F. Nagi, "A MATLAB based Face Recognition System using Image Processing and Neural Networks," *4th International Colloquium on Signal Processing and its Applications,* pp. 83-88, 2008.
[7]	A. Hailey, "DRIVER ALERTNESS DETECTION SYSTEM," Intercore, Inc., 2016. [Online]. Available: http://intercoreinc.com/. [Accessed October 2016].
[8]	M. A. Andrews, "The Sciences: Why do we yawn when we are tired? And why does it seem to be contagious?," 12 August 2002. [Online]. Available: https://www.scientificamerican.com/article/why-do-we-yawn-when-we-ar/. [Accessed 28 September 2016].
[9]	H. Joshua and R. Sally, "Fatigue Detection using Smartphones," *Ergonomics,* pp. 1-7, 2013.

[10]	J. Oliver, "Electroencephalogram (EEG)," Saint Luke's Health System, 2015. [Online]. Available: http://www.saintlukeshealthsystem.org/health-library/electroencephalogram-eeg. [Accessed 10 October 2016].
[11]	M. Teplan, "Fundamentals of EEG Measurement," *Measurement Science Review*, pp. 1-12, 2002.
[12]	"Fatigue and Rapid heart rate (pulse)," 2016. [Online]. Available: http://symptomchecker.webmd.com/multiple-symptoms?symptoms=fatigue%7Crapid-heart-rate-(pulse)&symptomids=98%7C184&locations=66%7C14. [Accessed 3 October 2016].
[13]	D. Vatta, R. Jindal and A. Gupta, "Face Detection using Digital Image Processing," *International Journal of Advanced Research in*, pp. 1-3, 2013.
[14]	Y.-Q. Wang, "An Analysis of the Viola-Jones Face Detection Algorithm," *Image Processing On Line*, p. 128–148, 2014.
[15]	Y. Mehta, "What is Internet of Things (IoT): with examples," Insticc, 20 November 2015. [Online]. Available: http://www.insticc.org/portal/NewsDetails/TabId/246/ArtMID/1130/ArticleID/738/What-is-Internet-of-Things-IoT-with-examples.aspx. [Accessed 15 October 2016].
[16]	R. Sam and H. Joshua, "Fatigue Detection using Smartphones," *Ergonomics*, pp. 1-7, 2013.
[17]	"Electroencephalogram (EEG)," 2015. [Online]. Available: http://www.saintlukeshealthsystem.org/health-library/electroencephalogram-eeg.
[18]	G. G. Schultz and H. T. Young, "A Safety Analysis of Fatigue and Drowsy Driving in the State of Utah," pp. ES-1, 2007.
[19]	J. Nagi, S. K. Ahmed and F. Nagi, "A MATLAB based Face Recognition System using Image Processing and Neural Networks," *4th International Colloquium on Signal Processing and its Applications*, pp. 83-88, 2008.

[20]	D. Liu, P. Sun, Y. Xiao and Y. Yin, "Drowsiness Detection Based on Eyelid Movement," Education, Technology and Computer Science (ETCS), 2010 Second International Workshop, pp. 49-52, 2010
[21]	M. A. Andrews, "The Sciences: Why do we yawn when we are tired? And why does it seem to be contagious?," 12 August 2002
[22]	B. B. Baharu, "Driver Drowsiness Detection by Using Webcam," Malaysia, 2013
[23]	F. Johannsson, R. Mollaei, J. Thor and J. Uusitalo, "GPS Satellite Signal Acquisition and Tracking", 21 August 1998