DROWSINESS DETECTION AND ALERTING SYSTEM

By

HARISH

DHARMADURAI

SUDHAN

ARULSUDHAN

PROJECT REPORT 1

Submitted to the naanmudhalvan smartinternz ibm related project

ABSTRACT

Drowsiness and fatigue are one of the main causes leading to road accidents. They can be prevented by taking effort to get enough sleep before driving, drink coffee or energy drink, or have a rest when the signs of drowsiness occur. The popular drowsiness detection method uses complex methods, such as EEG and ECG. This method has high accuracy for its measurement but it need to use contact measurement and it has many limitations on driver fatigue and drowsiness monitor [18]. Thus, it is not comfortable to be used in real time driving. This paper proposes a way to detect the drowsiness signs among drivers by measuring the eye closing rate and yawning.

This project describes on how to detect the eyes and mouth in a video recorded from the xperiment conducted by MIROS (Malaysian Institute of Road Safety). In the video, a participant will drive the driving simulation system and a webcam will be place in front of the driving simulator. The video will be recorded using the webcam to see the transition from awake to fatigue and finally, drowsy. The designed system deals with detecting the face area of the image captured from the video. The purpose of using the face area so it can narrow down to detect eyes and mouth within the face area. Once the face is found, the eyes and mouth are found by creating the eye for left and right eye detection and also mouth detection.

The parameters of the eyes and mouth detection are created within the face image. The video were change into images frames per second. From there, locating the eyes and mouth can be performed. Once the eyes are located, measuring the intensity changes in the eye area determine the eyes are open or closed.

If the eyes are found closed for 4 consecutive frames, it is confirm that the driver is in drowsiness condition.

Table of Contents

1. RESULT AND DISCUSSION
- FINALPROJECT DEVELOPMENT
- EXPERIMENTAL RESULT
- MODELING AND RESULT
- RESULT

⭕ APPENDIX

ABBREVIATIONS

ECG	Electrocardiography
EEG	Electroencephalography
FYP	Final Year Project

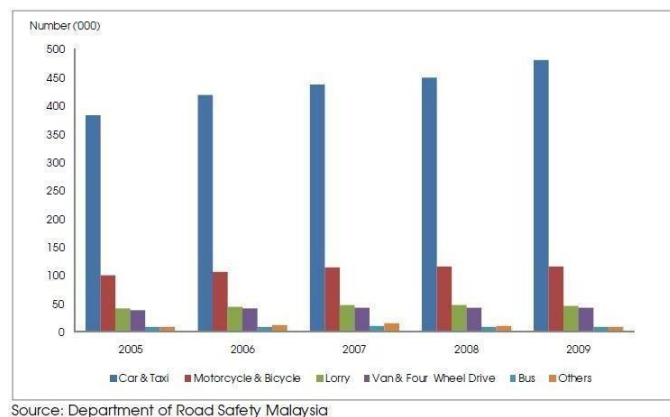MIROS   Malaysia Institute of Road Safety

PERCLOS   PERcentage of eye CLOSure

CHAPTER 1

INTRODUCTION

Drowsiness is a state of near sleep, where the person has a strong desire for sleep. It has two distinct meanings, referring both to the usual state preceding falling asleep and the chronic condition referring to being in that state independent of a daily rhythm [16]. Sleepiness can be dangerous when performing tasks that require constant concentration, such as driving a vehicle. When a person is sufficiently fatigue while driving, they will experience drowsiness and this leads to increase the factor of road accident.

Source: Department of Road Safety Malaysia

: Statistic of Road Accident from 2005 to 2009

Figure 1 shows the statistic of road accident in Malaysia from the year 2005 to 2009 provided by MIROS (Malaysia Institute of Road Safety). The numbers of vehicles involved in road accident keep increasing each year. From Figure 1, car and taxi type of vehicles shows about nearly 400,000 cases of road accident has been recorded. It keeps increasing every year and by the year 2009, it shows the number of road accident were recorded by MIROS are nearly 500,000.

shows the difference between fatigue and drowsiness condition.

Examples of Fatigue & Drowsiness Condition

The development of technologies for detecting or preventing drowsiness while driving is a major challenge in the field of accident avoidance system. Because of the hazard that drowsiness presents on the road, methods need to be developed for counteracting its affects.

The aim of this project is to develop a simulation of drowsiness detection system. The focus will be placed on designing a system that will accurately monitor the open or closed state of the driver's eyes and mouth. By monitoring the eyes, it is believed that the symptoms of driver's drowsiness can be detected in sufficiently early stage, to avoid a car accident. Yawning detection is a method to assess the driver's fatigue. When a person is fatigue, they keep yawning to ensure that there is enough oxygen for the brain consumption before going to drowsiness state . Detection of fatigue and drowsiness involves a sequence of images of a face, and the observation of eyes and mouth open or closed duration. Another method to detect eye closure is PERCLOS. This detection method is based on the time of eyes closed which refers to percentage of a specific time.

The analysis of face images is a popular research area with applications such as face recognition, and human identification and tracking for security systems. This project is focused on the localization of the eyes and mouth, which involves looking at the entire image of the face, and determining the position of the eyes and mouth, by applying the existing methods in image-processing algorithm. Once the position of the eyes is located, the system is designed to determine whether the eyes and mouth are opened or closed, and detect fatigue and drowsiness.

Background of Study

Each year, there is an increase in road accidents cases involving cars and heavy vehicles like buses, lorries and trucks in Malaysia. Drowsiness and fatigue condition is one of the prime factors contributing to road accidents. Driving in this condition may result terrible causes since it affects the driver's judgment and concentration. Falling asleep on the wheel can be avoid if the drivers

take efforts such as getting enough sleep before driving, taking caffeine or stop for a while to rest when the signs of fatigue and drowsiness appears.

However, in many cases, drivers refuse to take one of these steps even when they know that they are suffering from fatigue, and will continue driving. Therefore, detecting drowsiness is important as one of the steps to prevent the road accidents. This project proposed that yawning and eyes detection is the obvious signs of fatigue and drowsiness.

Significance of This Project

Drowsiness and fatigue lead the cause of road accident in Malaysia. Thus, Driver Drowsiness Detection by Using Webcam is being introduced to minimize and reduce the number of accidents involving cars, lorries and trucks. It detects the drowsiness signs and alerts drivers when they are in drowsy state.

Problem Statement

Current drowsiness detection systems monitoring the driver's condition requires complex computation and expensive equipment, not comfortable to wear during driving and is not suitable for driving conditions; for example, Electroencephalography (EEG) and Electrocardiography (ECG), i. e. detecting the brain frequency and measuring the rhythm of heart, respectively.

A drowsiness detection system which use a camera placed in front of the driver is more suitable to be use but the physical signs that will indicate drowsiness need to be located first in order to come up with a drowsiness detection algorithm that is reliable and accurate. Lighting intensity and while the driver tilt their face left or right are the problems occur during detection of eyes and mouth region.

Therefore, this project aims to analyze all the previous research and method, hence propose a method to detect drowsiness by using video or webcam. It analyzes the video images that have been recorded and come up with a system that can analyze each frame of the video.

Objectives
The project focuses on these objectives, which are:

- To suggest ways to detect fatigue and drowsiness while driving.

3

- To study on eyes and mouth from the video images of participants in the experiment of driving simulation conducted by MIROS that can be used as an indicator of fatigue and drowsiness.
- To investigate the physical changes of fatigue and drowsiness.
- To develop a system that use eyes closure and yawning as a way to detect fatigue and drowsiness.

Scope of Study

In this project, the author will focus on these following procedures:

- Basic concept of drowsiness detection system
- Familiarize with the signs of drowsiness
- Determine the drowsiness from these parameters
    - Eye blink
    - Area of the pupils detected at eyes
    - Yawning
- Data collection and measurement.
- Integration of the methods chosen.
- Coding development and testing.
- Complete testing and improvement.

Relevancy of the Project

This project is relevant to the implementation since fatigue and drowsiness drivers contribute to the percentage of road accidents. Many researches have been conducted to implement safe driving systems in order to reduce road accidents. Detecting the driver's alertness and drowsiness is an efficient way to prevent road accidents. With this system, drivers who are drowsy will be alerted by an alarm to regulate consciousness, attention and concentration of the drivers. This will help to reduce the number of road accidents.

This project is an active topic that is still being enhanced and improved by researches and can be applied in many areas such as detecting the attention-level of students in classrooms and lectures. This is also relevant to the three author's field of study since it requires the author to apply and combine the knowledge of electronics, programming and algorithms.

CHAPTER 2

IDEATION PHASE

PROPOSED SOLUTION:

PROBLEM STATEMENT:

Here We goes through some problem that face on accident due to drowsiness and tiredness. We had clear to provide drowsiness detection and alerting system to stop the accident.

Drowsiness describes the state of being drowsy or sleepy. A person experiencing drowsiness may feel tired or sluggish and have difficulty staying awake. Individuals are usually less alert when drowsy and may fall asleep but are able to be roused.

Nowadays, more and more professions require long-term concentration. Drivers must keep a close eye on the road, so they can react to sudden events immediately. Driver fatigue often becomes a direct cause of many traffic accidents. Therefore, there is a need to develop the systems that will detect and notify a driver of her/him bad psychophysical condition, which could significantly reduce the number of fatigue-related car accidents. However, the development of such systems encounters many difficulties related to fast and proper recognition of a driver's fatigue symptoms. One of the technical possibilities to implement driver drowsiness detection systems is to use the visionbased approach. This article presents the currently used driver drowsiness detection systems. Here we are detecting the driver drowsiness by estimating vision system of him. This paper presents an automatic drowsy driver monitoring.

Driver's inattention might be the result of a lack of alertness when driving due to driver drowsiness and distraction. Driver distraction occurs when an object or event draws a person's attention away from the driving task. Unlike driver distraction, driver drowsiness involves no triggering event but, instead, is characterized by a progressive withdrawal of attention from the road and traffic demands. Both driver drowsiness and distraction, however, might have the same effects, that is decreased driving performance, longer reaction time, and an increased risk of crash involvement.
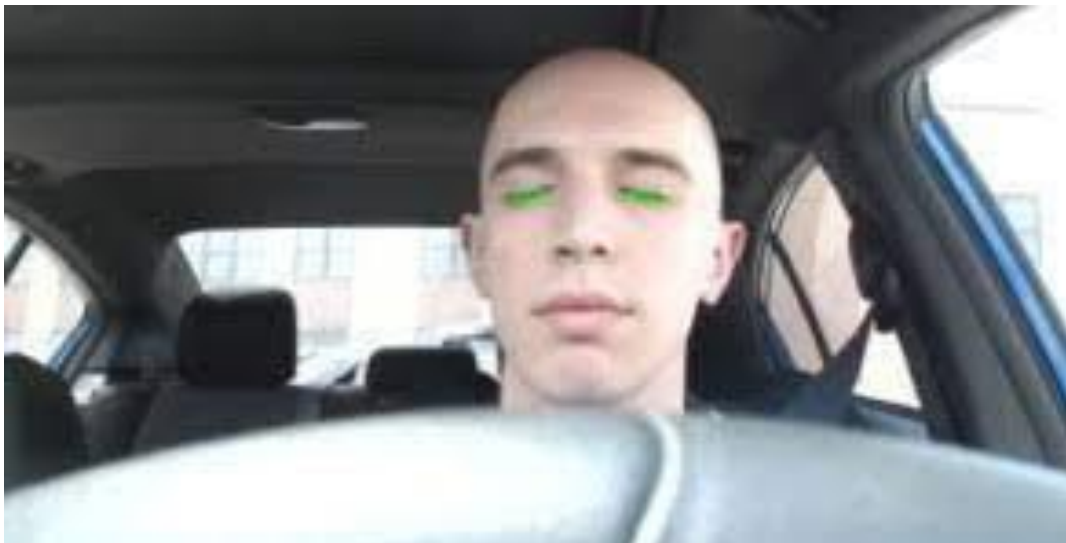
IDEA/SOLUTION DECRIPTION:

Based on acquisition of video from the camera that is in front of driver perform real-time processing of an incoming video stream in order to infer the driver's level of fatigue if the drowsiness is estimated then it will give the alert by sensing the eyes. A new approach towards automobile safety and security with autonomous region primarily based automatic automotive system is projected during this conception. A drowsy driver detection system and a traffic detection system with external vehicle intrusion dodging primarily based conception. So as to attenuate these problems, we've incorporated driver alert system by watching each the driver's eyes. Once its detected that the driver is drowsy then the particular score is generated and the alarm rings to make the driver aware.

GENERATE THE PREVIEW WITH THE WEB CAMERA .
RECORD THE VIDEO THROUGH THE WEB CAMERA AND CAPTURE IT
GENERATE THE PICTURES THROUGH THE VIDEO.

NOVELTY/UNIQUENESS:

The purpose of the drowsiness detection system is to aid in the prevention of accidents passenger and commercial vehicles. The system will detect the early symptoms of drowsiness before the driver has fully lost all attentiveness and warn the driver that they are no longer capable of operating the vehicle safely.
To determine whether the driver is drowsy, the study tests three ensemble algorithms and finds that the Boosting algorithm is the most effective in detecting drowsiness with an accuracy of 89.4%

SOCIAL IMPACT:

Drowsiness: Makes you less able to pay attention to the road. Slows your reaction time if you must brake or steer suddenly. Affects your ability to make good decisions.

Drowsy driving, also known as driver fatigue or tired driving, is the act of driving or operating a motor vehicle while tired and feeling fatigued or sleepy. Job stress or interrupted nights with young children are common reasons to be short on sleep.

However, other factors can contribute to drowsy driving, such as a medication you are taking or an untreated sleep disorder that leaves you depleted and unable to stay awake during the day. Late-night and third-shift workers are particularly affected by the natural release of melatonin associated with dark hours as they journey home after a long shift.

BUISENESS MODEL:

PERCLOS SystemsPERCLOS (percentage closure) is defined as the measurement of the percentage of time the pupils of the eyes are 80% or more occluded over a specified time interval. It has been found that PERCLOS is a reliable measure in detecting drowsiness.

The most popular algorithm for detecting drowsiness is PERCLOS. This algorithm is developed by Wierwille et al. PERCLOS measures percentage of time that eyes are closed over a window.

SCALABILITY OF THE SOLUTION:

Driver drowsiness detection systems can use cameras, eye tracking sensors and other hardware to monitor visual cues, where drowsiness can be detected through yawning frequency, eyeblinking frequency, eye-gaze movement, head movement and facial expressions

Eye tracking is the process of measuring either the point of gaze (where one is looking) or the motion of an eye relative to the head. An eye tracker is a device for measuring eye positions and eye movement. Eye trackers are used in research on the visual system, in psychology, in psycholinguistics, marketing, as an input device for humancomputer interaction, and in product design. Eye trackers are also being increasingly used for rehabilitative and assistive applications (related for instance to control of wheel chairs, robotic arms and prostheses). There are a number of methods for measuring eye movement. The most popular variant uses video images from which the eye position is extracted. Other methods use search coils or are based on the electrooculogram. We here are using eye tracking in detecting the drowsiness of the driver. Eye tracking is helping us to detect and sense the sleep of the driver, whether he is sleeping, wanting to sleep, getting exhausted while driving etc

Solution Architecture:

This phase uses Haar face detection algorithm that takes captured frames of image as input and then the detected face as output. Next, Haar is also used to extract the eyes image from the detected face which will be used as an input for the machine learning phase. The main role of the machine learning is to classify either the eyes of the driver are closed or opened using Support Vector Machine (SVM). If the result

of the classification indicates that the driver's eyes is closed for a predefined period of time, the eyes of the driver will be considered closed and hence an alarm will be started to alert the driver. The proposed methodology has been tested on available benchmark data. The result demonstrates the accuracy and robustness of the hybridized of image processing technique with machine learning technique. Thus, it can be concluded that the proposed approach is an effective solution method for a real-time of driver drowsiness detection.
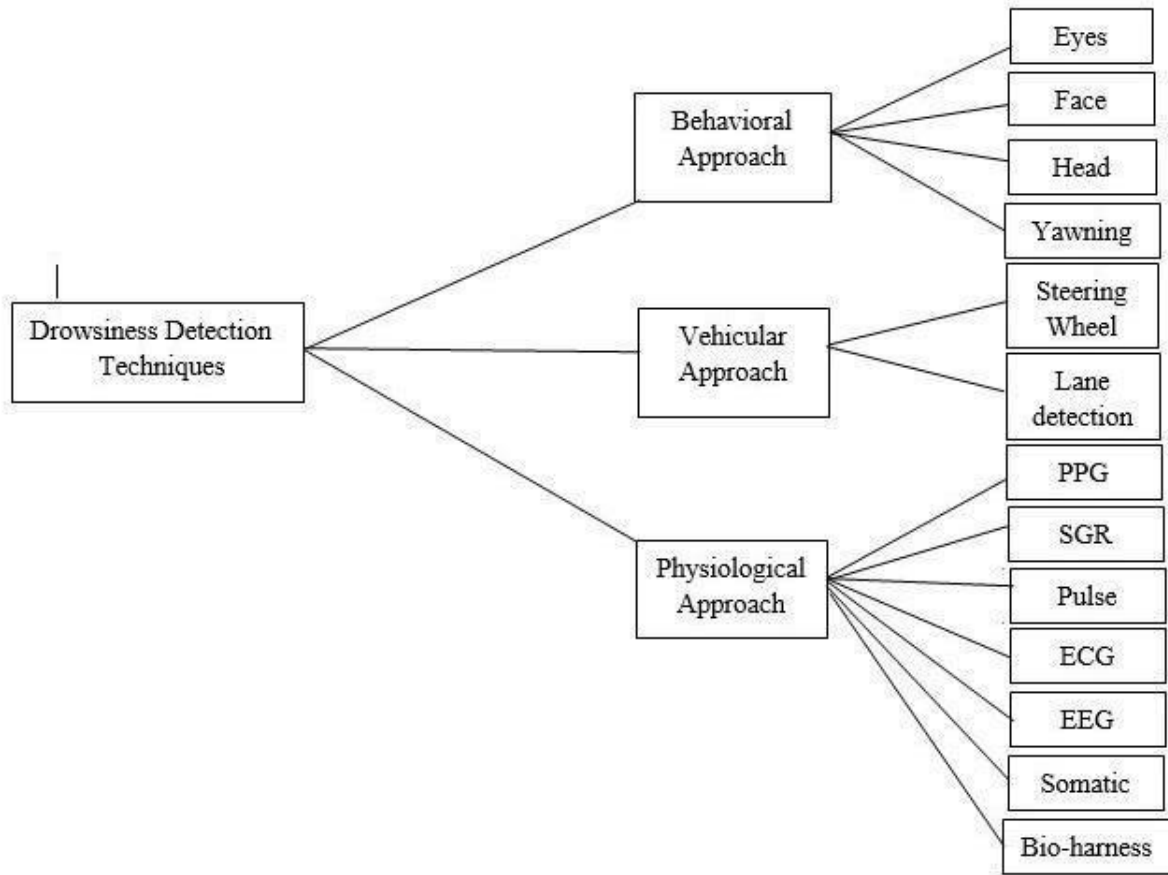
Driver drowsiness detection systems can use cameras, eye tracking sensors and other hardware to monitor visual cues, where drowsiness can be detected through yawning frequency, eye-blinking frequency, eye-gaze movement, head movement and facial expressions.

In an intrusive approach, sensors are used to detect driver drowsiness by placing them on the driver's body, whereas in a non-intrusive approach, a camera is used for drowsiness detection by identifying yawning patterns, eyelid movement and head inclination

The most popular algorithm for detecting drowsiness is PERCLOS. This algorithm is developed by Wierwille et al. [7]. PERCLOS measures percentage of time that eyes are closed over a window.

Popular algorithms used to perform object detection include convolutional neural networks (R-CNN, Region-Based Convolutional Neural Networks), Fast R-CNN, and YOLO (You Only Look Once).

EXAMPLE :

CHAPTER 3

There are many previous researches regarding driver drowsiness detection system that can be used as a reference to develop a real-time system on detecting drowsiness for drivers. There is also several method which use different approaches to detect the drowsiness signs. According to MIROS (Malaysia Institute of Road Safety), from the year of 2007 until 2010, they were 439 cases of road accidents have been investigated by the MIROS crash team .

Drowsiness and Fatigue

Antoine Picot *et al,* stated that drowsiness is where a person is in the middle of awake and sleepy state. This situation leads the driver to not giving full attention to their driving. Therefore, the vehicle can no longer be controlled due to the driver being in a semi-conscious state. According to Gianluca Borghini *et al,* mental fatigue is a factor of drowsiness and it caused the person who experiences to not be able to perform because it decreases the efficiency of the brain to respond towards sudden events.

Electroencephalography (EEG) for Drowsiness Detection

Examples of EEG Data Collecting [2]

Electroencephalography (EEG) is a method that measures the brain electrical activity. As shown in Figure , it can be used to measure the heartbeat, eye blink and even major physical movement such as head movement. It can be used on human or animal as subjects to get the brain activity. It uses a special hardware that place sensors around the top of the head area to sense any electrical brain activity.

Authors in mentioned that from the method that has been implemented by the previous researcher to detect drowsiness signs, the EEG method is best to be applied for drowsiness and fatigue detection. In the method, EEG have four types of frequency components that can be analyzed, i.e. alpha (α), beta (β), theta (θ) and delta (δ). When the power is increased in alpha (α) and delta (δ) frequency bands, it shows that the driver is facing fatigue and drowsiness .

The disadvantages of this method are, it is very sensitive to noise around the sensors. For example, when the person is doing the EEG experiment, the surrounding area must be completely silent. The noise will interfere with the sensors that detect the brain activity. Another disadvantage of this method is that even if the result might be accurate, it is not suitable to use for real driving application . Imagine when a person is driving and he is wearing something on his head with full

of wires and when the driver moves their head, the wire may strip off from their place. Even though it is not convenient to be used for real-time driving but for experiment purposes and data collection, it is one of the best methods so far .

Drowsiness detection using face detection system

Drowsiness can be detected by using face area detection . The methods to detect drowsiness within face area are vary due to drowsiness sign are more visible and clear to be detected at face area. From the face area, we can detect the eyes location. From eyes detection, author in stated that there are four types of eyelid movement that can be used for drowsiness detection. They are complete open, complete close, and in the middle where the eyes are from open to close and vice versa . Figure is an example of the image taken for detecting eyelid movement.

a) Open eye   b) Close eye   c) Processed close eye
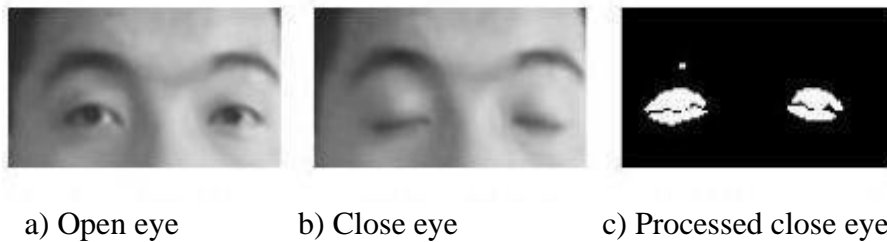
Examples of Eyelid Movement

The algorithm processes the images captured in grey-scale method; where the color from the images is then transformed into black and white. Working with black and white images is easier because only two parameters have to be measured. The author then performs the edge detection to detect the edges of eyes so that the value of eyelid area can be calculated.

The problem occurring with this method is that the size area of eye might vary from one person to another. Someone may have small eyes and looks like it is sleepy but some are not. Other than that, if the person is wearing glasses, there is obstacle to detect eye region.   The images that being captured must be in certain range from the camera because when the distance is far from the camera, the images are blurred .

PERCLOS (Percentage of Eye Closure)

Drowsiness can be captured by detecting the eye blinks and percentage of eye closure (PERCLOS). For eye blink detection, propose a method which learned the pattern of duration of

eyelid closed. According to , 'this proposed method measures the time for a person closed their eyes and if they are closed longer than the normal eye blink time, it is possible that the person is falling asleep'. In , the author mentioned that 'nearly 310.3ms are the average of normal person eye blink'.

PERCLOS method proposes that drowsiness is measured by calculating the percentage of the eyelid 'droops' . Sets of eye open and eye closed have been stored in the software library to be used as a parameter to differentiate either the eyes is fully open or fully closed. For eyelid to droops, it happened in much slower time as the person is slowly falling asleep. Hence, the transition of the driver's drowsy can be recorded. Thus, PERCLOS method put a proportional value where when the eyes is 80% closed, which it is nearly to fully close, it assumed that the driver is drowsy .

This method is not convenient to be used in real-time driving as it needs fix threshold value of eye opening for the PERCLOS method to perform accurately. Both methods to detect drowsiness using eye blink pattern and PERCLOS have the same problem where the camera need to be placed at a specific angle in order to get a good image of video with no disturbance of eyebrow and shadow that cover the eyes.

Yawning Detection Method

According to , drowsiness of a person can be observed by looking at their face and behavior. The author propose a method where drowsiness can be detected by mouth positioning and the images were process by using cascade of classifier that has been proposed by Viola-Jones for faces. The images were compared with the set of images data for mouth and yawning . Some people will close their mouth by their hand while yawning. It is an obstacle to get good images if a person is closing their mouth while yawning but yawning is definitely a sign of a person having drowsiness and fatigue. Figure are the examples of yawning detection method used in the research.

Examples of Person in Normal and Yawning Condition

After gone through the research papers and the existing methods, this project proposed that eyes and yawning detection method will be used. Eye blink duration gives the data that the longer the person's close their eyes, the drowsier it will be considered. It is because when a person is in drowsy state; its eyes will be closed longer than the normal eye blink. Other than that, yawning is one of the symptoms of drowsiness where it is a normal human response when yawning is the sign that they feel drowsy or fatigue.

CHAPTER 3

METHODOLOGY

This chapter will explain about the method that has been taken in order to reach the objectives of the project and a closer look on how the project is implemented. It is the analysis of each stage that will be faced in order to complete this project. Each selection and achievement of the method taken that has been implement in this project will be explained for each stage until the project is success. This project involves software usage which is MATLAB® Computer Vision System. The methods used are existing method in MATLAB® command to detect face, eyes, and mouth area.

Research Methodology

Usually, research methodology refers to a set of procedures that will be used to carry out a certain research. In order to complete this project systematically within the specified time, there are some methodologies and activities that need to be planned and followed consistently.

Flow Chart
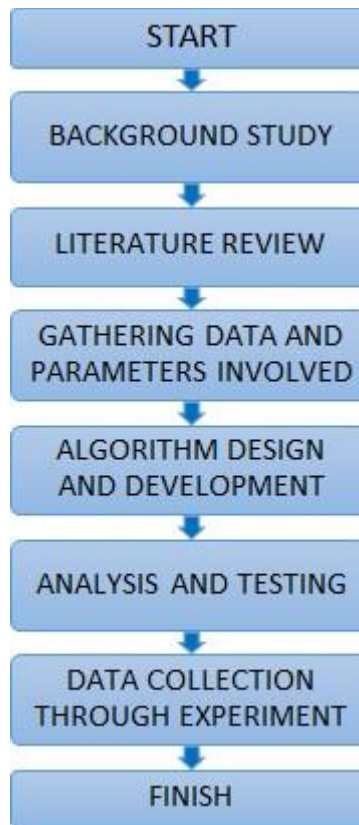
Flow Chart of Project Progress

Background of Study

Before starting any research or project, basic information of the related topic is required to ensure that the author understands what the project is all about. In this stage, the background of study helps the author understands the relation between drowsiness and fatigue. It also helps the author in understanding the seriousness of driving a motored vehicle in drowsiness condition. It is proven that driving the vehicle in fatigue and drowsiness condition is a lead factor to road accidents.

Literature Review

In this stage, it involved the study of the previous research done related to author project. This topic observes the relationship between drowsiness condition and handling a motored vehicle. A thorough observation was done on the existing method to detect the drowsiness. Different

parameters have been used by previous researches. By focusing on the parameters which is detecting eyes and mouth, helps to narrow down the perspective of this project.

Previous Data Gathering and Analysis

In this stage, it was found that one of the best way to detect eyes and yawning is by algorithm. Some of current algorithms that are related to this project are reviewed to help developing the project. In [10], the propose method measures the time for a person closed its eyes and if its eyes are closed longer than the normal eye blink time, it is possible that the person is falling asleep. Based on researches of human eye blinks, it has been identified that the average of human blink duration takes about 202.24ms while the blink duration of a drowsy individual takes about 258.57ms.

After defining the method to be used in this project, authors obtain the video of the experiment conducted by MIROS where participants are driving in a simulated environment and being recorded for the whole session. The experiment takes time, about 60 to 90 minutes. The analysis of drowsiness detection is done manually by watching the full length of the video and mark down the drowsiness signs. The parameters of the data are: drowsiness, yawing and other signs occurs at starting and ending time. This is to calculate the duration of the signs occurred.

Algorithm Design and Development

A few algorithms and technique has been used in the process of detecting face, eyes and mouth. The algorithm and technique used is Cascade Object Detector. The Cascade Object Detector uses the Viola-Jones algorithm to detect people's face, nose, eyes, mouth or upper body.

Viola-Jones Face Detection Algorithm

Viola-Jones object detection framework can be used to detect a variety of object classes, but it is more focused on the detection of face and facial features. This algorithm uses the concept of rectangle features which involves the sums of pixels within the rectangular areas. From Figure the sums of the pixels that lie within the white rectangles are subtracted from the sum of pixels in the grey rectangles.

The value of a *two-rectangle feature,* which are represented by A and B is the difference between the sum of pixels within two rectangular regions. The regions have the same size and shape. They

are also horizontally or vertically oriented and adjacent to each other. A *three-rectangle feature,* which is represented as C, computes the sum within two outside rectangles subtracted from the sum in a center rectangle. Finally, a *four-rectangle feature,* which is represented as D computes the difference between diagonal pairs of rectangles.
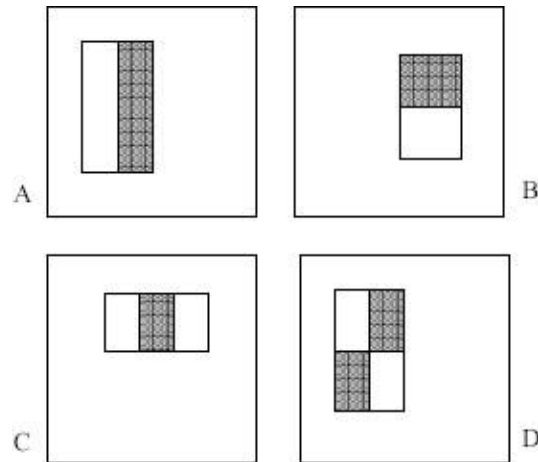
Feature Used in Viola-Jones

Rectangle features can be computed rapidly using a representation for the image which is called integral image.
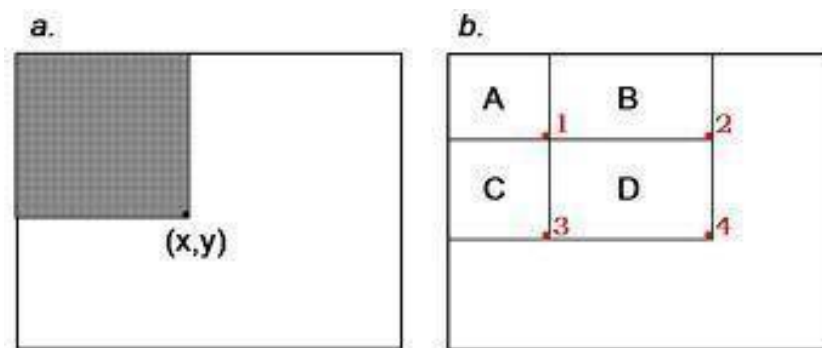

Integral Image

The value of the integral image at point (x, y) is the sum of all the pixels above and to the left. Based on the integral image, the sum of the pixels within rectangle D can be computed with four

array references. The value of the integral image at location 1 is the sum of the pixels in rectangle A. The value at location 2 is A + B, at location 3 is A + C, and at location 4 is A + B + C + D.

For the task of face detection, the features that are selected to detect the location of the face are shown in Figure .The two features are shown in the top row and are overlay on a typical training face in the bottom row. The first feature measures the difference in intensity between the region of the eyes and a region across the upper cheeks. This is because the region of the eyes is often darker than the region of the nose and upper cheeks. The second feature compares the intensities in the eye regions to the intensity across the bridge of the nose.
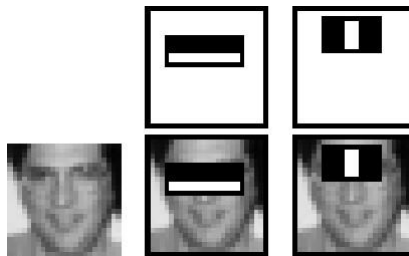
Feature for Face Detection

Cascade of Classifiers

In a standard 24x24 pixel sub-window, there are a total of 45,396 possible features that are detected. This is a number that is too large and prohibitively expensive to be evaluated. In order to improve detection performance, features need to be added to the classifiers. However, this step directly increases the computation time and making the detection process much slower. Therefore, a cascade of classifiers is constructed to increase detection performance while radically reducing computation time.
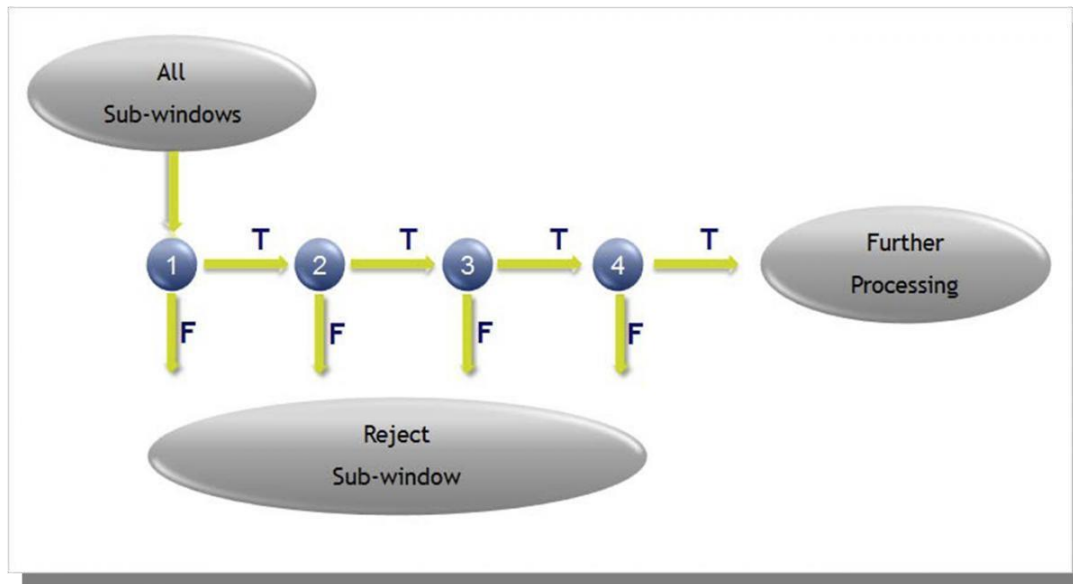
Cascade of Classifiers

The evaluation of the strong classifiers generated by the learning process can be done quickly but it is not fast enough to run in real-time. Therefore, the strong classifiers are arranged in a cascade in order of complexity. Each successive classifier is trained only on selected samples which pass through the preceding classifiers. If at any stage in the cascade a classifier rejects the sub- window under inspection, no further processing is performed and it continues to search the next sub-window.

Analysis and Testing

In order to detect eye and mouth area, the face area need to be detected first. However, this step will reduce the performance and the speed of the system due to a large area of detection. The project objective is the detection of the drowsiness signs which is eyes and mouth area. Therefore, this project limited the area of detection to eye and mouth. This will enhance the performance of the system. Cascade Object Detector algorithm is being tested by using MATLAB® software to get the detection area for the system that will be develop. Testing need to be done to ensure it fulfils the required parameters needed.

Gantt Chart

Table 1: Gantt chart of Final Year Project I & II

No	Activities	Weeks													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
	FYP 1 Progress and Milestone (May 2013)														
1	Title selection	■	■												
2	Preliminary Research & Literature Review		■	■	■	■	■								
3	Methods Identification				■	■			■	■	■	■			
4	Familiarize with the Method and techniques								■	■	■	■	■	■	■
5	Video Analysis								■	■	■	■			
	FYP II Progress and Milestone (September 2013)														
1	Data collection and Measurement.	■	■												
2	Integration of the Methods chosen		■	■	■										
3	Coding Development and Simulation				■	■	■	■	■	■	■	■			
4	Complete simulation and Improvement							■	■	■	■	■	■	■	■
5	Testing and Modification			■	■	■	■	■	■	■	■	■	■	■	■
6	Report / Thesis			■	■	■	■	■	■	■	■	■	■	■	■

Final Project I Milestones:

- Completion of title selection
- Completion of preliminary research & literature review
- Completion of methods identification
- Completion of understanding the techniques and methods
- Analyzing video experiment

Final Project II Key Milestones:

- Data Collection & Measurement
- Integrations Of Methods Chosen
- Coding Development And Simulation   • Complete Simulation & Improvement
- Testing And Modification

Tools

A few tools have been used to detect the face and further extract the eye and mouth region.

Logitech C310 web camera

- Used as a tool to capture images of the driver's face before narrowing it down to the eye region and the mouth.

Video Cutter

- Used as a tool to cut the video from the experiment conducted by MIROS. The video need to be cut to be used in MATLAB® software.
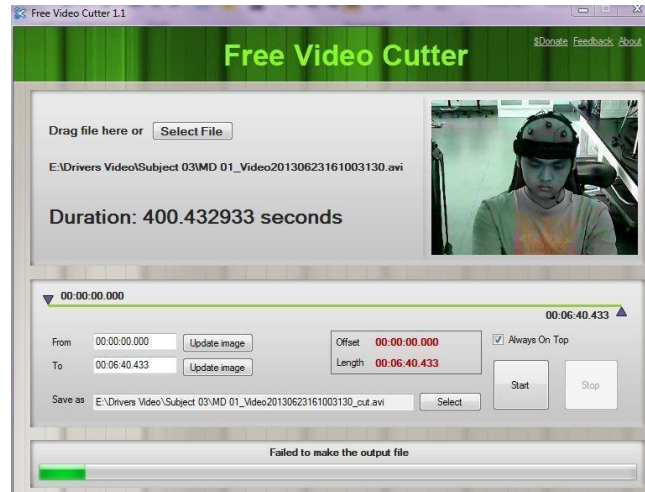
Video Cutter Software

Format Factory Software

- Used to converts video by joining multiple video file to get one video.
- Used to converts the video format to suit the MATLAB® requirement. • Compatible to convert any video file formats.
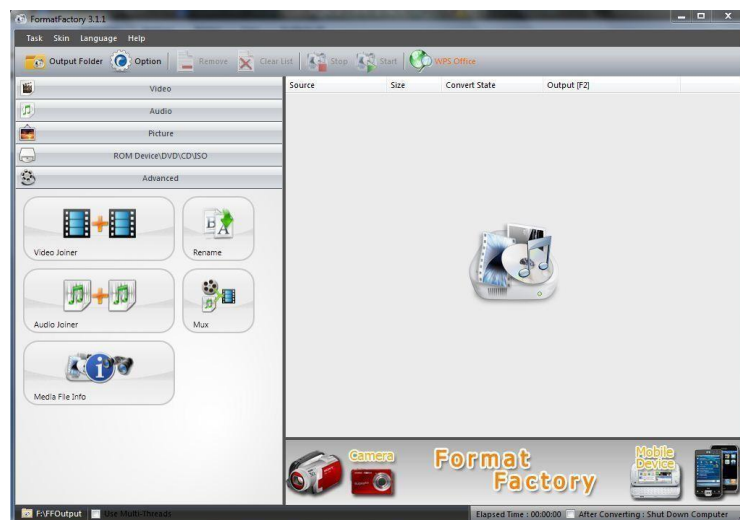
MATLAB® R2013a Software

- Used as one of the means to implement algorithm for eye blink detection system.
- Compatible for various programming languages such as C, C++, and Java.
- Computer Vision System Toolbox is used to provide algorithms and tools for the design and simulation of computer vision and video processing. The system includes algorithms for feature extraction, motion detection, object tracking, video processing and video analysis.
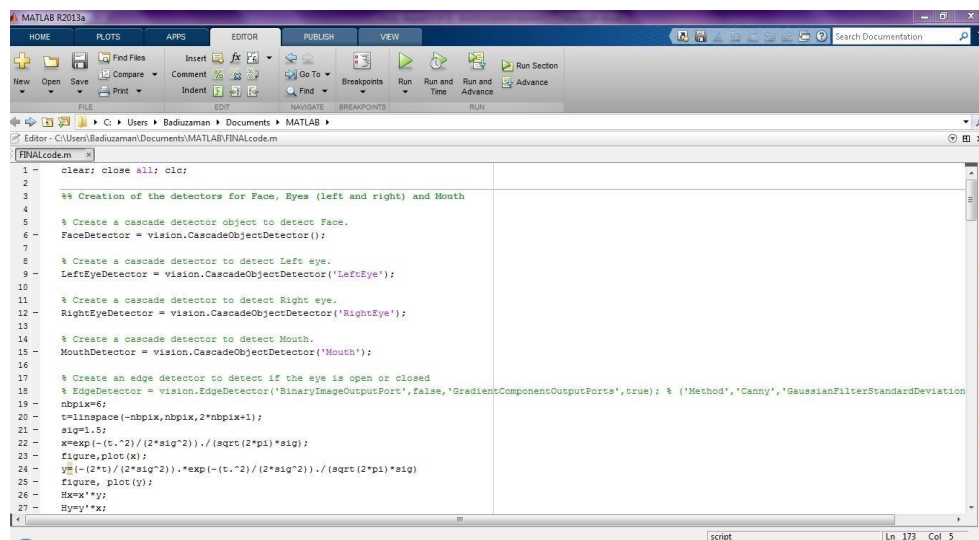
Example of MATLAB® Interface

CHAPTER 4

RESULT AND DISCUSSION

## Final Project II Development

This chapter will discuss about the result obtained related to the final year project. Other than that, this chapter analyses the information related to the algorithm simulation. The steps of detecting eyes and mouth also will be explained in this chapter.

Experimentation Result

In earlier stages of this final year project, it has been intended to use MATLAB® to develop the algorithm for the system. The remaining of FYP 1 time period and during the FYP 2 period has been scheduled for working on the coding for the system. Using MATLAB®, toolboxes of the software have been used to ease the process of writing the codes. In the experimentation result, detecting face, eyes and mouth has been done.
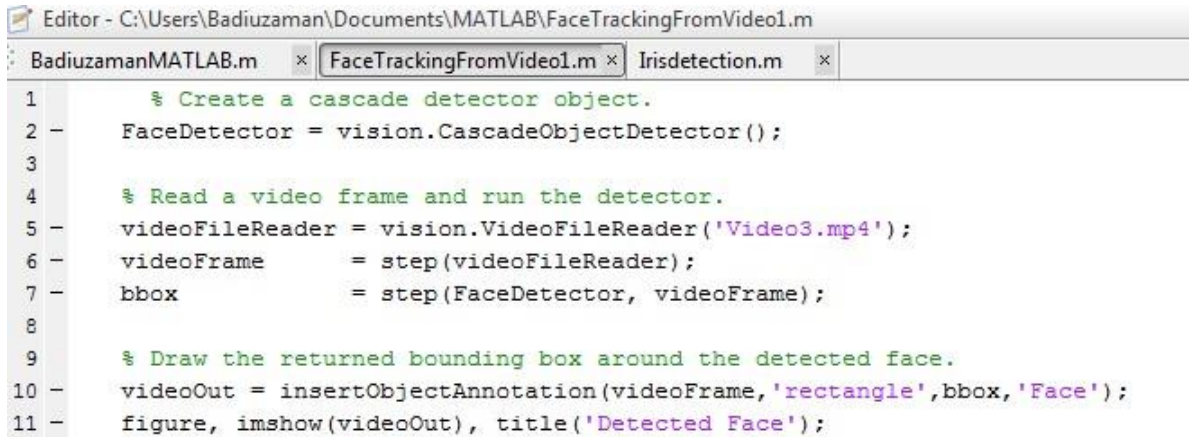
Detecting Face

To detect the face, author use the algorithm that are part of the Computer Vision Toolbox System which is Vision Cascade Detector. It creates a system object detector that detects object using Viola-Jones method. By default, the detector is configured to detect faces. Figure shows the command script and the result of the face detection algorithm.
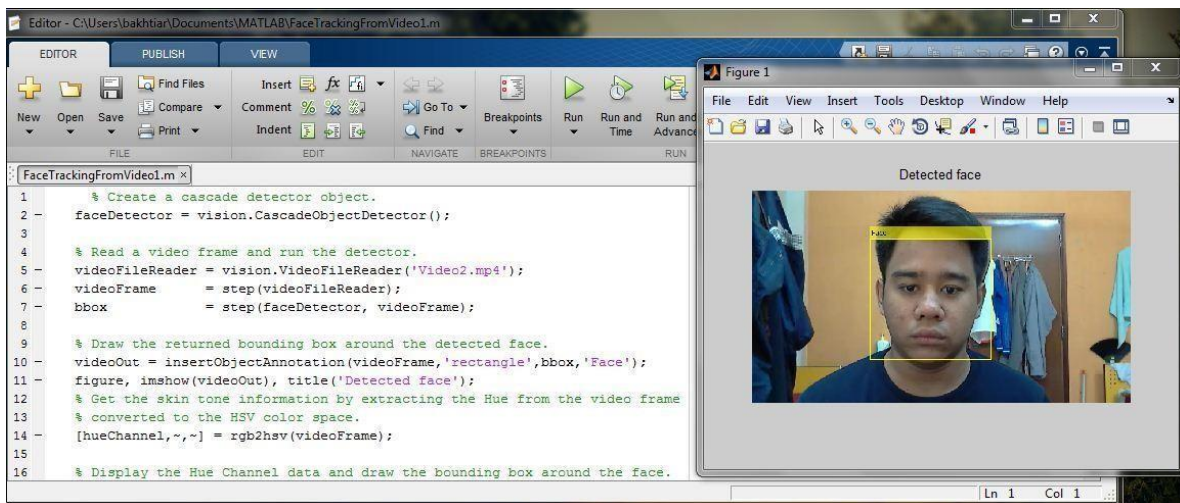
Face Detection Result

The features to detect face area are as below:

1. Define and setup the cascade object detector using the constructor. The constructor uses built-in Viola-Jones algorithm to detect faces, noses, eyes, mouth and upper body.
2. Read the video or the image selected and run the face detector.
3. Draw the bounding box around the detected face. The bounding box is the area of the desired detected face. Here, the bounding box is around the face area. Figure shows the command from the MATLAB® editor.

```
Editor - C:\Users\Badiuzaman\Documents\MATLAB\FaceTrackingFromVideo1.m

BadiuzamanMATLAB.m    ×  FaceTrackingFromVideo1.m ×  Irisdetection.m    ×

 1           % Create a cascade detector object.
 2 -    FaceDetector = vision.CascadeObjectDetector();
 3
 4           % Read a video frame and run the detector.
 5 -    videoFileReader = vision.VideoFileReader('Video3.mp4');
 6 -    videoFrame      = step(videoFileReader);
 7 -    bbox            = step(FaceDetector, videoFrame);
 8
 9           % Draw the returned bounding box around the detected face.
10 -    videoOut = insertObjectAnnotation(videoFrame,'rectangle',bbox,'Face');
11 -    figure, imshow(videoOut), title('Detected Face');
```

Command from MATLAB® Editor.

Detecting Eyes

The eyes must be detected separately because when the drivers tilt their face, it still can be detected. The author uses the same algorithm as the face detection but here it changes the object to detect eyes. Figure shows the command from the MATLAB® editor. The resulting output of the eye detection algorithm is shown in Figure . When author tries using different videos, a problem occurs; other parts in the video are detected as eyes. If the detected region is within the eye area, it is consider as True Positive. But during the trial, other parts in the video were detected as eyes far from the eye region which it is considered as False Positive. Definition for False Positive is a result that is erroneously positive when a situation is normal. The results of False Positive using other video is shown in Figure .

```
Editor - C:\Users\Badiuzaman\Documents\MATLAB\BadiuzamanMATLAB.m

BadiuzamanMATLAB.m  ×   FaceTrackingFromVideo1.m  ×   Irisdetection.m  ×

1         % Create a cascade detector object.
2  -      faceDetector = vision.CascadeObjectDetector();
3
4         % Create a cascade detector to detect Left eye.
5  -      LeftEyeDetector = vision.CascadeObjectDetector('LeftEye');
6
7         % Create a cascade detector to detect Right eye.
8  -      RightEyeDetector = vision.CascadeObjectDetector('RightEye');
9
10        % Read a video frame and run the detector.
11 -      videoFileReader = vision.VideoFileReader('video5.mp4');
12 -      videoFrame      = step(videoFileReader);
13 -      bbox            = step(faceDetector,videoFrame);
14
15 -      cgface=[bbox(1,1)+bbox(1,3)/2, bbox(1,2)+bbox(1,4)/2];
16
17 -      videoFileReader = vision.VideoFileReader('video5.mp4');
18 -      videoFrame      = step(videoFileReader);
19 -      bbox_1          = step(LeftEyeDetector,videoFrame);
20
21        % Remove outlier left eye
22 -      [nl,nc]=size(bbox_1);
23 -  ┌   for i=1:nl
24 -  │       cgtmp=[bbox_1(i,1)+bbox_1(i,3)/2, bbox_1(i,2)+bbox_1(i,4)/2 ];
25 -  │       if( (cgtmp(1,1)<cgface(1,1)) && (cgtmp(1,2)<cgface(1,2) ) )
26 -  │           CoordLeftEye=bbox_1(i,:);
27 -  │       end
28 -  └   end
```

Fi Command to Detect Eyes


24

Result of Eyes Detection
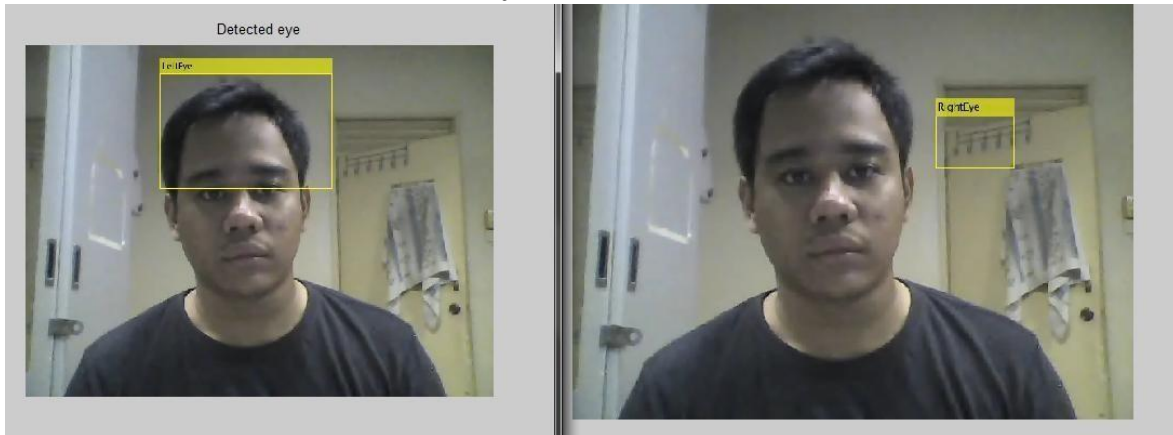
False Negative Eyes Detection

Detecting Mouth

Detecting mouth objective is to detect the drowsiness symptom which is yawning. To detect mouth, the cascade object detector that use Viola-Jones algorithm has been used by detecting object in rectangle shape.

The features to detect mouth area are as below:

4. Define and setup the cascade object detector using the constructor. The constructor uses built-in Viola-Jones algorithm to detect faces, noses, eyes, mouth and upper body.
5. Read the video or the image selected and run the face detector.
6. Draw the bounding box around the detected mouth. The bounding box is the area of the desired detection. But the algorithm to detect mouth is not as efficient as it detects other parts in the video as mouth. Figure shows the algorithm from the MATLAB® editor.
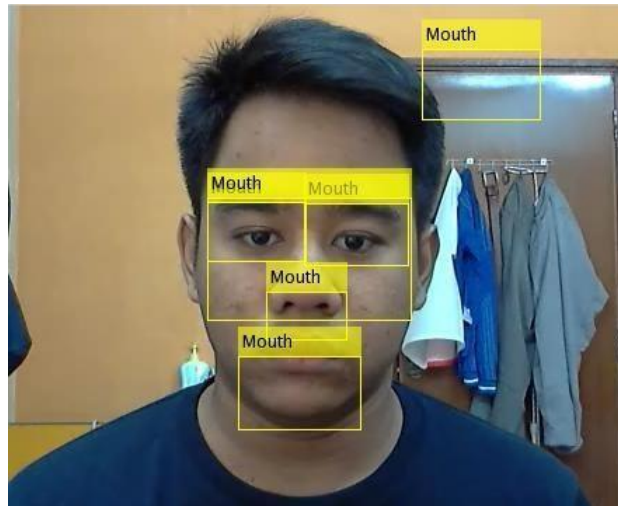
Fail Negative Mouth Detection

Figure shows the algorithm to detect mouth. The command need to be improved to make sure that it can detect the desired part which is mouth area by removing the wrong bounding boxes.


```
Editor - C:\Users\Badiuzaman\Documents\MATLAB\mouthdetection.m
BadiuzamanMATLAB.m*   ×   FaceTrackingFromVideo1.m ×   Irisdetection.m    ×   mouthdetection.m   ×
1         % Create a cascade detector object.
2 -     Detector = vision.CascadeObjectDetector('Mouth');
3
4        % Read a video frame and run the detector.
5 -     videoFileReader = vision.VideoFileReader('Video2.mp4');
6 -     videoFrame      = step(videoFileReader);
7 -     bbox            = step(Detector,videoFrame);
8
9        % Draw the returned bounding box around the detected face.
10 -    videoOut = insertObjectAnnotation(videoFrame,'rectangle',bbox,'Mouth');
11 -    figure, imshow(videoOut), title('Detected Mouth');
```

Command to Detect Mouth

Modelling and Result

Modelling

After the algorithm has been developed during the experimentation part, it needs improvement for the system meet the objectives of the project. Figure below shows the flow process of the improved

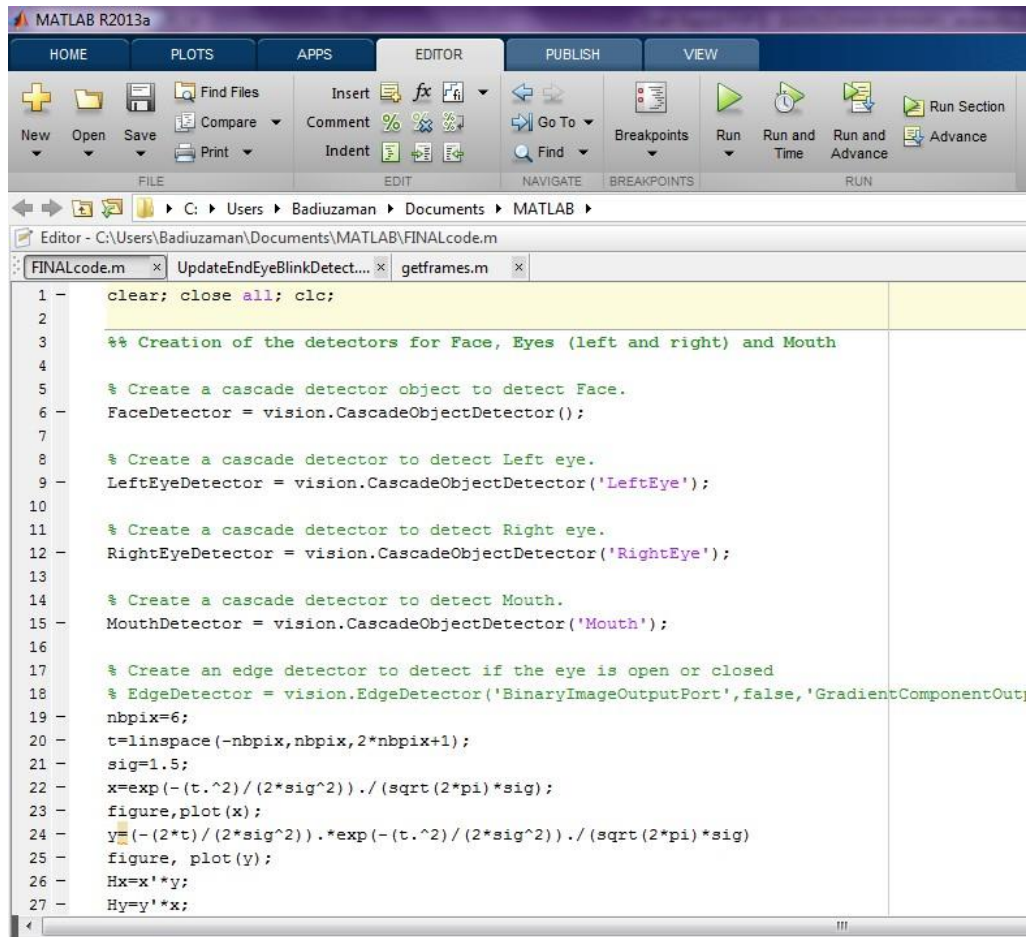algorithm that has been developed. The algorithm successfully detects the eyes and mouth in the video; hence the result can be obtained.
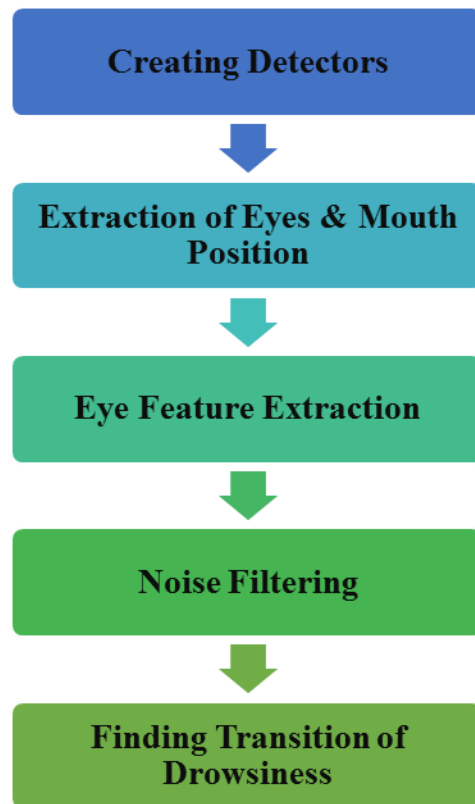
Flow Process of Detection Algorithm

Creating Detectors

In the first section, all the cascade detectors for face, each eye and mouth were created. It is because the entire algorithm was related to the other detectors feature. Figure below shows the detectors algorithm.

```
1 -   clear; close all; clc;
2
3     %% Creation of the detectors for Face, Eyes (left and right) and Mouth
4
5     % Create a cascade detector object to detect Face.
6 -   FaceDetector = vision.CascadeObjectDetector();
7
8     % Create a cascade detector to detect Left eye.
9 -   LeftEyeDetector = vision.CascadeObjectDetector('LeftEye');
10
11    % Create a cascade detector to detect Right eye.
12 -  RightEyeDetector = vision.CascadeObjectDetector('RightEye');
13
14    % Create a cascade detector to detect Mouth.
15 -  MouthDetector = vision.CascadeObjectDetector('Mouth');
16
17    % Create an edge detector to detect if the eye is open or closed
18    % EdgeDetector = vision.EdgeDetector('BinaryImageOutputPort',false,'GradientComponentOutp
19 -  nbpix=6;
20 -  t=linspace(-nbpix,nbpix,2*nbpix+1);
21 -  sig=1.5;
22 -  x=exp(-(t.^2)/(2*sig^2))./(sqrt(2*pi)*sig);
23 -  figure,plot(x);
24 -  y=(-(2*t)/(2*sig^2)).*exp(-(t.^2)/(2*sig^2))./(sqrt(2*pi)*sig)
25 -  figure, plot(y);
26 -  Hx=x'*y;
27 -  Hy=y'*x;
```

Cascade Detectors for Face, Each Eye and Mouth.
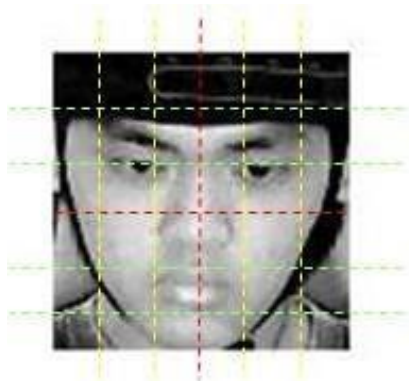
Extraction of Eyes & Mouth Position

After creating the detectors, bounding box of face, each eye and mouth areas is obtained. Each bounding box contains the parameters of the detection part. Cascade filters were used to extract the detected facial features by getting the centre gravity of the face bounding box. It is obtain by dividing the face bounding box (x, y) value by 2 and marked by red dash-line in the image in Figure 24. To require the True Positive detection, the centre gravity of face bounding box (x, y) value is used and it is divided by 3. Figure show the algorithm to obtain the centre gravity of the face.
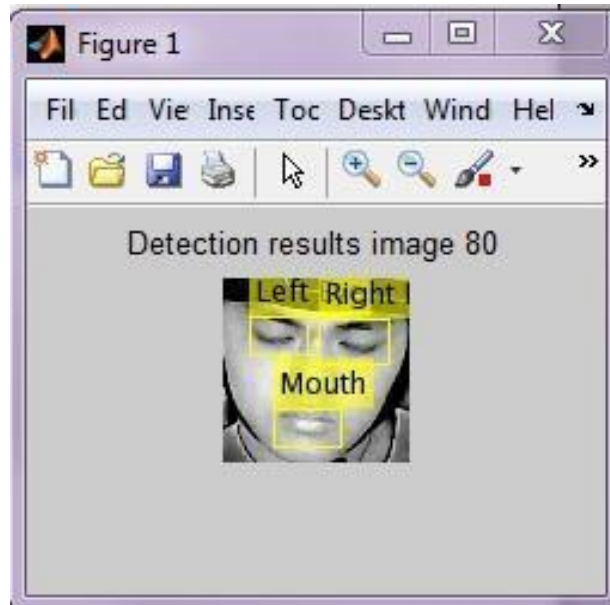
Centre Gravity of Face Detection Algorithm.

Cascade Filters of Face Image

The extraction is done to ensure the eyes detection bounding box are at the upper part of face and the mouth detection bounding box is below nose or at lower part of face. Figure 26 shows the True Positive detection obtained.

True Positive Eyes and Mouth Detection.

Eye Feature Extraction

After getting the bounding box of each eye, histogram equalization filter is applied to overcome the luminance variation problem. Each grayscale image which the value of each pixel is single sample, carry only the intensity information where black at the weakest intensity and white at the strongest. Luminance intensity is a measure of wavelength-weight power emitted by a light source in a particular direction per unit solid angle. Histogram equalization is a technique for adjusting image intensities to enhance contrast.

It seems that the features are directly related to eye closing. It is assumed that the amount of gradient in eye image is proportional to the opening of the eyes. For example, if a person has eyes opened, the quantity of gradient might be superior to the quantity of gradient when eyes are closed. Here, a detection of cumulative quantity in time region will inform us on the opening or closing of the eye. Figure shows the algorithm to perform the histogram equalization.
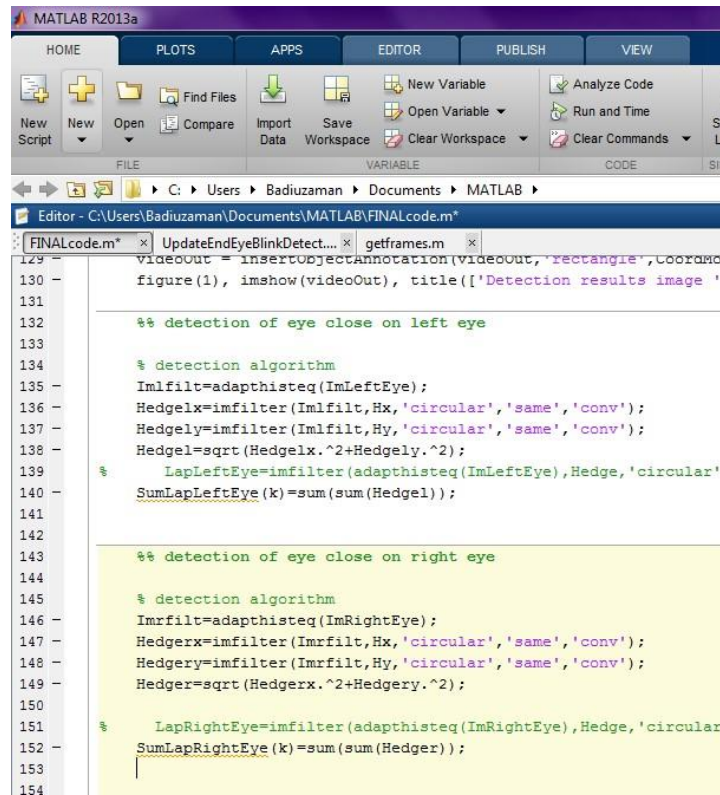
Histogram Equalization Filter Algorithm

After developing the algorithm to overcome the luminance problems, the luminance images were convolve with Derivative of Gaussian (DoG). The convolution is performed to get the gradient of the images by summing the gradient of the images value; the quantity of the edges for each eye can be obtained.

Noise Filter

For data collection, median filter is used in the algorithm to filter the noise for the final graph is smooth. After that, global parameter method is used to get the transition from eye open to eye close, done by the product energy of each region. Lastly, the global parameter is derived by using DoG. The same methods are used in the algorithm to detect the opening of the mouth.

Figure shows the full process of the drowsiness detection system.
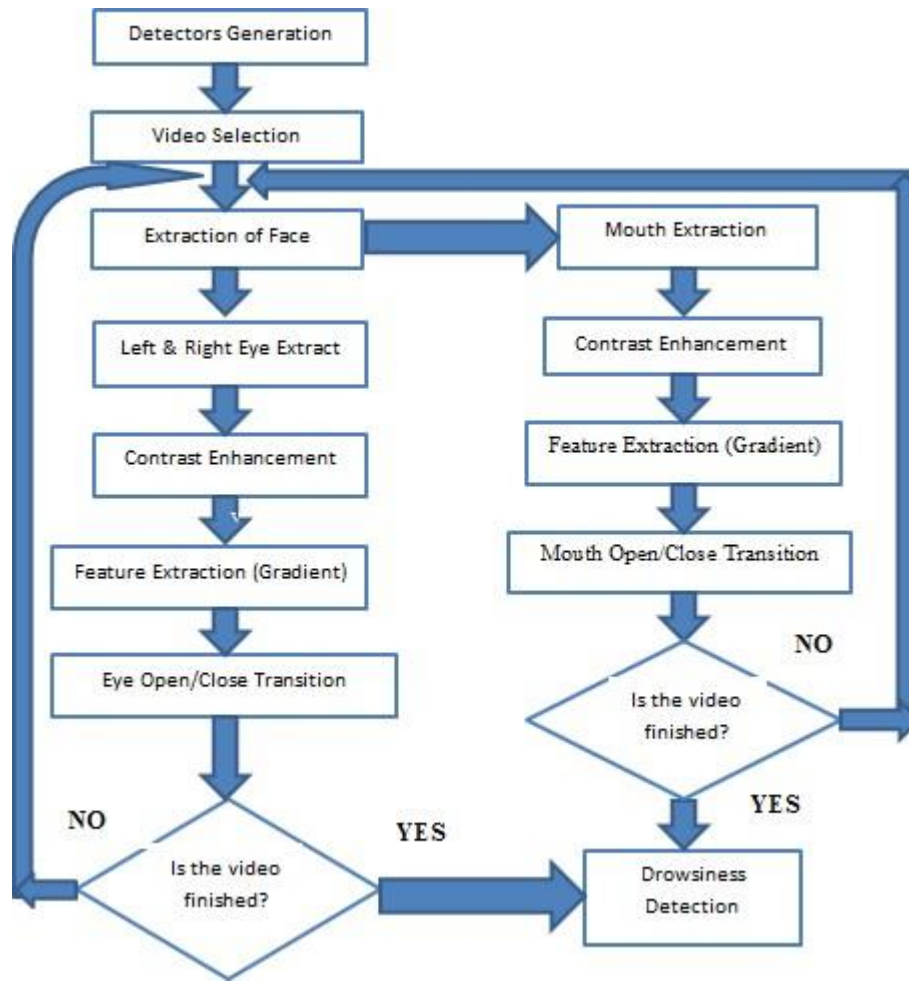
: Flow Process of Drowsiness Detection System

Results

After completing the algorithm modelling, the simulation of algorithm was performed by using the video from the experiment conduct by MIROS. Due to the video length cannot be run in MATLAB® because the processor have not enough memory, the video were separated into 3 minutes each video. By doing that, each video contains not more than 400 images per video. There a 10 participants who complete the driving experiment but as to show in the result, we only take a participant. Total video that have been separate is 31 videos.

The peak value in the graph show the driver is in drowsiness state.

The driver is drowsy because the driver closed his eyes for more than 4 frames per second.

33
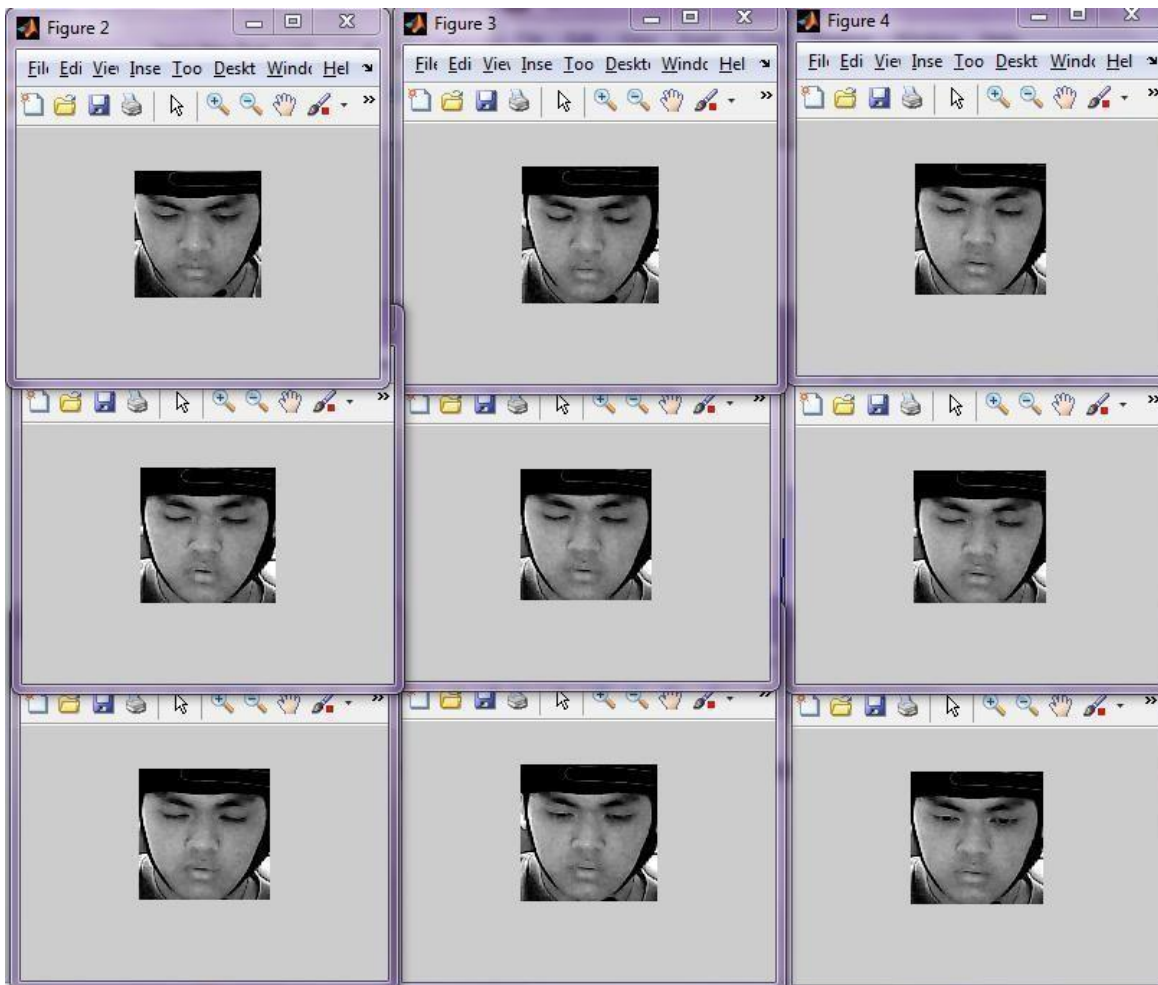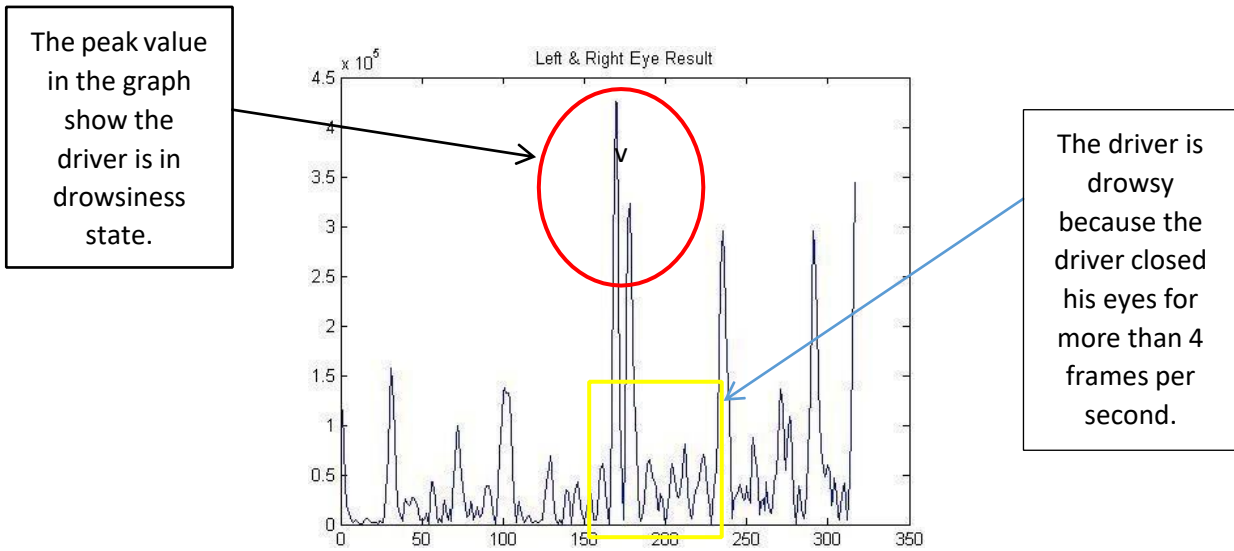
In the image above, the drowsiness state occur from frame number 168 to 179, where the driver closed it eyes for 11 seconds. The result of the driver yawns just after the drowsiness state occurs and it is shown in the graph .
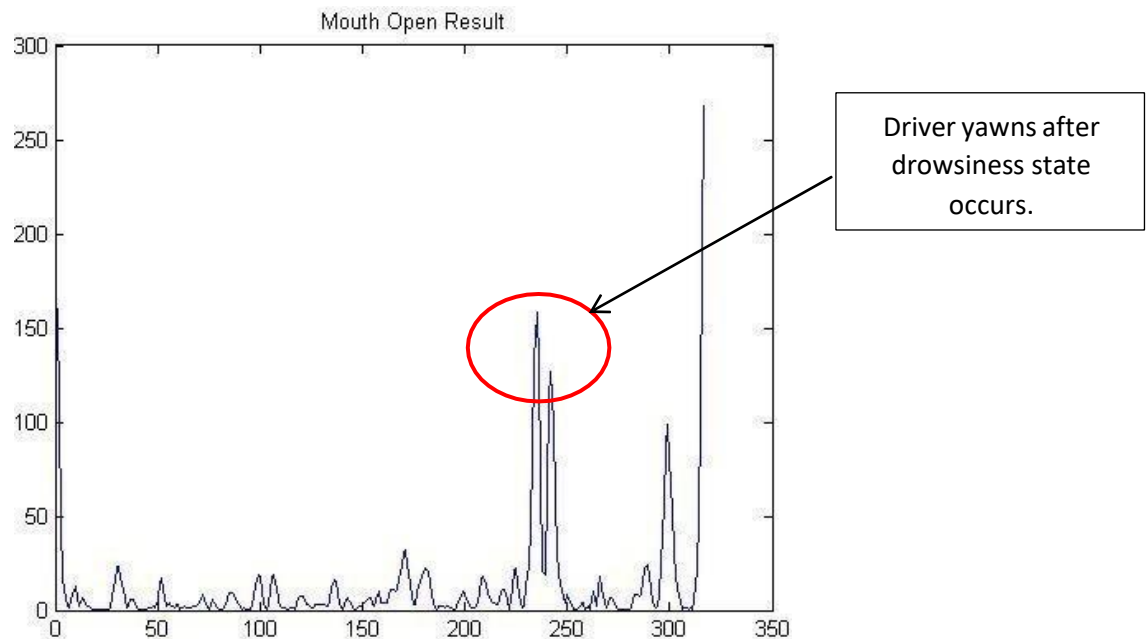
From the experimentation and modelling, this project successfully detects each eyes and mouth for drowsiness detection system to meet the objectives of Final Year Project. This system is reliable because it detects the transition of the drowsiness state where the algorithms detects each eyes and mouth changes each frame per second. Therefore, the Drowsiness Detection System by using Webcam has successfully done by using MATLAB® Software. All the data and result from the simulation algorithm has been shown this chapter.

APPENDIX

Source code detection.py

```
from scipy.spatial import distance
from imutils import face_utils
import imutils import dlib import
cv2
```

```python
def eye_aspect_ratio(eye):
A = distance.euclidean(eye[1], eye[5])
B = distance.euclidean(eye[2], eye[4])    C = distance.euclidean(eye[0], eye[3])
        ear = (A + B) / (2.0 * C)
        return ear


thresh = 0.25 frame_check
= 20
detect = dlib.get_frontal_face_detector()
predict = dlib.shape_predictor("models/shape_predictor_68_face_landmarks.dat")# Dat file is the crux of the code


(lStart, lEnd) = face_utils.FACIAL_LANDMARKS_68_IDXS["left_eye"] (rStart,
rEnd) = face_utils.FACIAL_LANDMARKS_68_IDXS["right_eye"]
cap=cv2.VideoCapture(0)
flag=0 while
True:
        ret, frame=cap.read()
        frame = imutils.resize(frame, width=450)          gray =
cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        subjects = detect(gray, 0)
for subject in subjects:
                shape = predict(gray, subject)
                shape = face_utils.shape_to_np(shape)#converting to NumPy Array
                leftEye = shape[lStart:lEnd]
rightEye = shape[rStart:rEnd]              leftEAR =
eye_aspect_ratio(leftEye)                  rightEAR =
eye_aspect_ratio(rightEye)                 ear = (leftEAR +
rightEAR) / 2.0               leftEyeHull =
cv2.convexHull(leftEye)                  rightEyeHull =
cv2.convexHull(rightEye)
                cv2.drawContours(frame, [leftEyeHull], -1, (0, 255, 0), 1)
        cv2.drawContours(frame, [rightEyeHull], -1, (0, 255, 0), 1)
if ear < thresh:
  flag += 1       print (flag)       if flag >=
frame_check:
                        cv2.putText(frame, "*****ALERT!*****", (10, 30),
cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)                        cv2.putText(frame,
"*****ALERT!*****", (10,325),                        cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0,
255), 2)
                        #print ("Drowsy")
        else:
                flag = 0
        cv2.imshow("Frame", frame)
key = cv2.waitKey(1) & 0xFF    if key
== ord("q"):
                break
cv2.destroyAllWindows()
cap.release()
```

DROWSINESS DETECTION jpnb:

```
{
 "cells": [
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "# Drowsiness Detection OpenCV\n",
    "\n",
    "\n",
    "This code can detect your eyes and alert when the user is drowsy.\n",
    "\n",
    "## Applications\n",
    "This can be used by riders who tend to drive for a longer period of time that may lead to accidents.\n",
    "\n",
    "### Algorithm\n",
    "\n",
    "Each eye is represented by 6 (x, y)-coordinates, starting at the left-corner of the eye (as if you were looking at the person), and then working clockwise around the eye:.\n",
    "\n",
    "<img src=\"eye1.jpg\">\n",
    "\n",
    "### Condition\n",
    "\n",
    "It checks 20 consecutive frames and if the Eye Aspect ratio is lesst than 0.25, Alert is generated.\n",
    "\n",
    "#### Relationship\n",
    "\n",
    "<img src=\"eye2.png\">\n",
    "\n",
    "#### Summing up\n",
    "\n",
    "<img src=\"eye3.jpg\">\n",
    "\n",
    "jupyter kernel by Manuel Romero (mrm8488@gmail.com or @mrm8488)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 1,
   "metadata": {},
   "outputs": [],
   "source": [
```

```
   "from scipy.spatial import distance\n",
   "from imutils import face_utils\n",
   "import imutils\n",
   "import dlib\n",
   "import cv2"
 ]
},
{
 "cell_type": "code",
 "execution_count": 2,
 "metadata": {},
 "outputs": [],
 "source": [
  "def eye_aspect_ratio(eye):\n",
  "\tA = distance.euclidean(eye[1], eye[5])\n",
  "\tB = distance.euclidean(eye[2], eye[4])\n",
  "\tC = distance.euclidean(eye[0], eye[3])\n",
  "\tear = (A + B) / (2.0 * C)\n",
  "\treturn ear"
 ]
},
{
 "cell_type": "code",
 "execution_count": 3,
 "metadata": {},
 "outputs": [],
 "source": [
  "thresh = 0.25\n",
  "frame_check = 20\n",
  "detect = dlib.get_frontal_face_detector()\n",
  "predict = dlib.shape_predictor(\"shape_predictor_68_face_landmarks.dat\")"
 ]
},
{
 "cell_type": "code",
 "execution_count": 4,
 "metadata": {},
 "outputs": [],
 "source": [
  "(lStart, lEnd) = face_utils.FACIAL_LANDMARKS_IDXS[\"left_eye\"]\n",
  "(rStart, rEnd) = face_utils.FACIAL_LANDMARKS_IDXS[\"right_eye\"]"
 ]
},
{
 "cell_type": "code",
 "execution_count": 5,
 "metadata": {},
 "outputs": [],
```

```
  "source": [
   "cap=cv2.VideoCapture(0)\n",
   "flag=0\n",
   "while True:\n",
   "\tret, frame=cap.read()\n",
   "\tframe = imutils.resize(frame, width=450)\n",
   "\tgray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)\n",
   "\tsubjects = detect(gray, 0)\n",
   "\tfor subject in subjects:\n",
   "\t\tshape = predict(gray, subject)\n",
   "\t\tshape = face_utils.shape_to_np(shape)#converting to NumPy Array\n",
   "\t\tleftEye = shape[lStart:lEnd]\n",
   "\t\trightEye = shape[rStart:rEnd]\n",
   "\t\tleftEAR = eye_aspect_ratio(leftEye)\n",
   "\t\trightEAR = eye_aspect_ratio(rightEye)\n",
   "\t\tear = (leftEAR + rightEAR) / 2.0\n",
   "\t\tleftEyeHull = cv2.convexHull(leftEye)\n",
   "\t\trightEyeHull = cv2.convexHull(rightEye)\n",
   "\t\tcv2.drawContours(frame, [leftEyeHull], -1, (0, 255, 0), 1)\n",
   "\t\tcv2.drawContours(frame, [rightEyeHull], -1, (0, 255, 0), 1)\n",
   "\t\tif ear < thresh:\n",
   "\t\t\tflag += 1\n",
   "\t\t\t#print (flag)\n",
   "\t\t\tif flag >= frame_check:\n",
   "\t\t\t\tcv2.putText(frame, \"*****ALERT!*****\", (10, 30),\n",
   "\t\t\t\tcv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)\n",
   "\t\t\t\tcv2.putText(frame, \"*****ALERT!*****\", (10,325),\n",
   "\t\t\t\tcv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)\n",
   "\t\telse:\n",
   "\t\t\tflag = 0\n",
   "\tcv2.imshow(\"Frame\", frame)\n",
   "\tkey = cv2.waitKey(1) & 0xFF\n",
   "\tif key == ord(\"q\"):\n",
   "\t\tcv2.destroyAllWindows()\n",
   "\t\tcap.release()\n",
   "\t\tbreak\n"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "metadata": {},
  "outputs": [],
  "source": []
 }
],
"metadata": {
 "kernelspec": {
```

```
   "display_name": "Python 3",
   "language": "python",
   "name": "python3"
  },
  "language_info": {
   "codemirror_mode": {
    "name": "ipython",
    "version": 3
   },
   "file_extension": ".py",
   "mimetype": "text/x-python",
   "name": "python",
   "nbconvert_exporter": "python",
   "pygments_lexer": "ipython3",
   "version": "3.6.4"
  }
 },
 "nbformat": 4,
 "nbformat_minor": 2
}
```

The above codes are the source code and detection.py and detection jpnb

The codes that are run in the python simulator to executes the code.

CONCLUSION

Previously, the author focuses on developing the algorithm or command to detect drowsiness. The developments of the algorithm takes time due to the authors only have basic skill in using MATLAB®. The author learns about the MATLAB® commands by developing the algorithm only with the help from Computer Vision Toolbox System that is already built-in in MATLAB® software and also by trial and error of the shared file from the MathWorks®. MathWorks® is where all the high skills of MATLAB® users from all over the world share their works on algorithms.

Other than that, during the Final Year Project I, investigating the drowsiness signs and collecting the data from the video of the experiments have been the main job scope. It will be used as

parameters to develop the simulation system in detecting drowsiness. Until now, one of this semester's project objectives has been achieved which is to study the video images of participants in the experiment of driving simulation conducted by MIROS. Several techniques to develop the simulation system have been discovered. There are also other objectives that this project needs to achieve.

For the progress of Final Year Project II, the author started developing the algorithm to detect the drowsiness. Few techniques have been implemented in this project which has been found through the previous researches. Further adjustment of the algorithm and the techniques need to be done in order to meet the requirement of this project and to finish it within the given time frame.

Videolink: https://github.com/naanmudhalvan-SI/IBM--17668-1682488548/tree/ae3de4f8b345d552feb26c19b4f0db2585a2d59e/demo video