SMART DRIVING SYSTEM

BY

GC ABDUL HANNAN BABAR

GC FAISAL WAQAR

GC HAMZA JASPAL

GC HARIS SALEEM

Supervisor

Maj Ajlaan Bin Mamoon

Submitted to the Faculty of Electrical Engineering Department, Military College of Signals, National University of Sciences and Technology, Islamabad

In partial fulfillment for the requirements of a B.E Degree in Electrical Engineering

May 2019

CERTIFICATE OF APPROVAL AND CORRECTNESS

This is to officially state that the thesis work contained in this report "Smart Driving System" Is carried out by GC Abdul Hannan Babar, GC Faisal Waqar, GC Hamza Jaspal and GC Haris Saleem under my supervision and that in my judgment, it is fully ample, in scope and excellence, for the degree of Bachelor of Electrical Engineering from Military College of Signals, National University of Sciences and Technology (NUST). And is original with 14 % of plagiarism.

Approved By:

Signature: __________________________

Supervisor: Maj Ajlaan Bin Mamoon

MCS, Rawalpindi

DECLARATION OF ORIGINALITY

We hereby declare that the work contained in this report and the intellectual content of this report are the product of our work. This thesis report has not been formerly published in any structure nor does it include any verbatim of the published resources which could be treated as violation of the international copyright decree. We also affirm that we do recognize the terms ‘plagiarism’ and ‘copyright’ and that in case of any copyright infringement or plagiarism established in this thesis, we will be held fully accountable of the consequences of any such violation.

DEDICATION

In the name Allah, the Most Merciful, the Most Beneficent.

To our parents, without whose unflinching support and unstinting cooperation, a work of this magnitude would not have been possible.

ACKNOWLEDGEMENTS

In the name of Allah the most gracious, the most benevolent. All praise to Him who enabled us to work on this project and by His grace we were able to take it to task. We would also extend our deepest gratitude to our supervisor Maj Ajlaan Bin Mamoon without whom this work would not have been possible. We would also like to thank our instructors who were available to us at all times when ever we needed their esteemed guidance.

ABSTRACT

This project introduces a smart driving system which will be specifically used for military purposes. It is supposed to monitor a number of aspects of driving that includes driver's drowsiness, vehicle speed and time constraints with respect to the use of vehicles as per SOP (Standard Operating Procedures). The monitoring will be real time making alarms and prior notifications in case of misuse/unfortunate incidents as well as data which will be communicated to the concerned authorities.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

1 Introduction

Road accidents are very common in daily life. Every morning we find such bad news and they are just increasing with passage of time. Numerous reasons are there behind these casualties mostly including abiding traffic rules. Sleep is a natural process but during driving it causes serious issues even deaths. Military is all about discipline in all its field including transport as well. For military drivers it is necessary to follow all MT (military transport) Standard Operating Procedure to achieve their target. Many times, due to hard routine and sleepless routine it become difficult for them to drive easily.

Introducing Smart Driving System which is supposed to monitor a number of aspects of driving that includes driver's drowsiness, vehicle speed, distance covered and time constraints with respect to the use of vehicles as per SOP (Standard Operating Procedures). The monitoring will be real time making alarms and prior notifications in case of misuse/unfortunate incidents as well as real time monitoring which will be communicated to the concerned authorities on an HTML web page accessed by the IP of the MCU.

1.1 Problem Statement

Human error can often cause great damage to life and property. Moreover, human resource is a very expensive aspect especially when it comes to Military. It is an intrinsic characteristic of a human being to feel sleepy or drowsy anytime, anywhere. This natural flaw can be very dangerous when it comes to accidents and can even cost their valuable lives. It has become necessary to apply such device in a vehicle to stop them from falling asleep and to alert them if the driver falls asleep.

1.2 Background

Development of advancements to unendingly screen the condition of the driver is basic all together. To give reasonable administrations to arranged driving circumstances. For as long as decades an assortment of driver state observing strategies have been proposed from numerous investigations. Driver readiness state observing frameworks would by and large work dependent on driving examples and driver's physiological sign. Driver's driving examples are helpful to obtain, yet to evaluate driver state precisely is a difficult errand on the grounds that these techniques survey the driver state by implication. Then again, investigation dependent on driver's physiological sign has for quite some time been done to screen the condition of the driver straightforwardly, however the sensors were not favored because of the sensor's low convenience in vehicle climate. It was noticed that sluggish drivers were repeatedly making mistakes in their discipline. Rather, there was a previous time of quantifiable execution decrement with associated psycho physiological signs. Tiredness can be identified with practical accuracy utilizing driving execution measures for example, directing and changes in vehicle extraneous path position. Numerous endeavors have been accounted for in the writing for building up a functioning security framework for lessening the quantity of cars mishaps because of decreased readiness.

Among available techniques to counter driving issues following methods are being used

1. The use of real time face recognition techniques for the detection of a driver's facial features.
2. Real time data collection, processing and uploading on a web portal to enable us observe and monitor the results.
3. Providing an on-the-go and easily available solution to monitor real time speed and distance and integrating it all on Micro Controller Unit and using Wi-Fi module to communicate the readings to the servers.

1.3 Project Scope

The scope of our smart driving system is to solve the excessive deployment of Military Police in Cantonments for the purpose of monitoring military vehicles. Remove human errors and misuse of assets against SOPs. System will be able to prevent a number of accidents that results in both financial and human life loss.

1.4 Project Specifications

Advancement of an easy to use and financially applicable model by mixing computerized picture handling and control framework fields to lower the bad consequences. It very well may be conveyed in genuine vehicular framework to keep the driver alert.

The development of a smart driving system includes

1. Drowsiness detection which will be done by image processing [1].
2. Detection of human face and recognition of important facial marks such as eyes. This can be done by the use of Voila Jones algorithm [2] which processes image to see where lies the eyes of humans.
3. Further the distance is calculated between the eyes and the eye lids.
4. Now with the information about the distance between the eyelids of open eyes comparison can be done at real time with a distance where the eyes are closed.
5. If this condition is true for a specific period of time the driver will be declared sleeping and the alarm will sound to alert the driver.
6. There will be rotary encoders and sonar sensors making reading of the vehicles speed and monitoring if the car is getting close to anything on either side.
7. This information will be given as input to the Arduino Due to process it.
8. Using the Wi-Fi module this will be communicated to the concerned authorities. An html page will be made to display this information to the

designated supervisor which will be accessed by the IP address associated to the Wi-Fi module of the MCU.

1.5 Project Deliverables

The Smart Driving System will be implemented using Voila Jones algorithm [2] along with MCU that is being used for monitoring of the vehicle. Providing with all possible requirements it will be much easier for concern authorities to keep an eye on the driver and his activities. This project will give following deliverables:

1. Protect driver from falling asleep.
2. Stopping driver from over speeding.
3. Keeping his record.
4. Allowing authorities to monitor the vehicle anytime.
5. Reducing fatalities.
6. All possible at low price.

Organization of Thesis

Chapter 2: This chapter contains the literature review and research work which is required for image processing leading to drowsiness detection.

Chapter 3: This chapter contains the detailed development and working of the entire project from image processing to calculation of speed of a moving vehicle and the transferring of data over the internet.

Chapter 4: This chapter has the critical analysis of our projects design in aspect to its feasibility, user friendliness and universal compatibility.

Chapter 5: This chapter lays all the conclusion leading to the end of our project.

Chapter 6: The last chapter of the thesis throws light on all the future aspects related to this work.

Chapter 2

2 Literature Review

2.1 Digital Image

An image is defined as 2-dimensional functions f(x,y), where x and y are called as spatial coordinates and the amplitude of these coordinates (x,y) at any pair is known as grey level or intensity of the image at that point. When the x,y and amplitude values are all discrete quantities, finite, we call the image a digital image. Digital image is all about processing of digital images by using digital computers. Digital image is made up of elements each of has a particular value and location. These elements are known as image elements, pixels or picture elements [3].

2.2 Digital Image Processing

Performing image processing [1] on digital images using digital computer algorithms is known as digital image processing. Moreover, as a sub-field of digital signal processing, digital image processing has as well as so many advantages over analog image processing.

Digital image processing mainly focuses on the following tasks

1. Improvement of digital pictorial representation for human interpretation.
2. Processing of digital image for representation, transmission and storage for uncontrolled machine perception.

Table 1: Levels of Image Processing

Low Level Process	**Mid-Level Process**	**High Level Process**
Input: Image	Input: Image	Input: Attributes
Output: Image	Output: Attributes	Output: Understanding
Examples: Noise removal, Image sharpening	Examples: Object recognition, segmentation	Examples: Scene understanding, Autonomous navigation

In our project the development of a smart driving system includes drowsiness detection which will be done by image processing. The base of this is detection of human face and recognition of important facial marks such as eyes. This can be done by the use of Voila Jones algorithm [2] which processes image to see where lies the eyes of humans.

2.3 Voila Jones Algorithm

Every image is made up of pixels. These pixels can be converted into mathematical models using computational techniques and various algorithms. These algorithms and process combine up to form what is known as digital image processing

Following are its major processes:

1. Signals are converted from sensors to digital image.
2. Quality of the image is improved by removing noise.

3. Major aspects/features are extracted out.
4. Image is prepared for display and printing.
5. Images can also be compressed to be transferred.

In 2002 Paul Viola Michael Jones came up with the paper named as “Rapid Object Detection using a Boosted Cascade of a Simple Features”. Despite the fact that deep learning has taken over everything. In face detection, this still performs absolutely fine. It’s incredibly quick and any kind of camera that does some kind of face detection, it’s going to use something very similar to this.

There are some few problems with the face detection one is that we don’t know how big face is going to be. It could be very big could be very small, and another is, you have got a very high-resolution image. We want to be doing this lots and lots of times a second. So, what are going to do? Look over every tiny bit of image lots and lots of times? Machine language says it’s a face or it’s not a face. There’s a trade-off between speed and accuracy and false-positives and false-negatives. We have people from different walks of life. Youth, elders, people with spectacles, things like this. So, all of this adds up to a quite a problem, and yet it’s not a problem. These people came up with a classifier that uses very simple features, one bit of an image subtracted from another bit of an image.

This feature doesn’t look for the facial features like nose eyes etc rather it uses edges and other features and it combines them together into objects using deep learning. For example, if we are just looking at a grayscale image, eye is slightly darker than forehead.

The features they use is very simple. So, they are just using one of the features that is two rectangle features for the detection of faces. So, we have got two rectangles next to each other which, are some number of pixels like 9 pixels each or maybe hundred pixels each and we subtract one from other. So essentially, we are looking for bits of an image where one bit is darker or brighter than another bit.

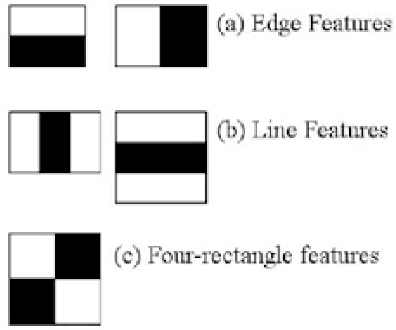


Figure 2-1: Image Detection through rectangle features [3]

In above figure. Edge feature as 2 rectangle features. Line as 3 rectangle feature and (c) as 4 rectangle features.

Calculating large groups of pixels and summing them up is quite a slow process. So, they introduced an idea called an integral image which makes this way faster. Let's imagine we have an image and consider we want to calculate these bits of images. Imagine we have an image which is nice and small. After arithmetic calculations that is every new pixel is the sum of all pixels above and to the left and it including it. So, this would be something like this

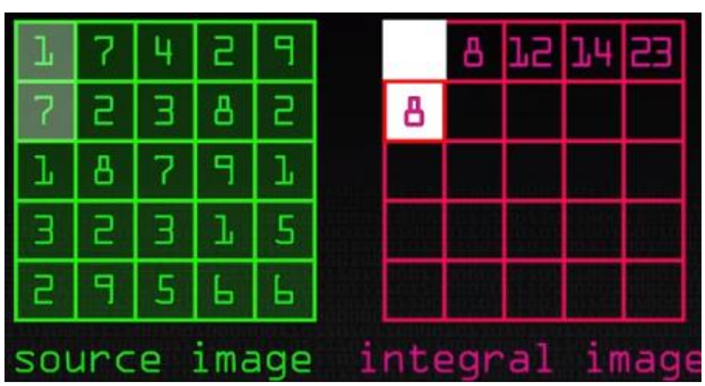


Figure 2-2: Source to integral image [3]

So, what they do, is calculate all of the features, right for a 24 by 24 image they calculate all 180,000 possible combinations of 2,3 and 4 rectangle features and they work out which one for a given data set of faces and not faces, which one best separates the positive from the negatives. So, let's say you have 10000 pictures of faces 10,000 pictures of background which one feature best says "this is a face this is not a face" right bearing in mind nothing is going to get it completely right with just one feature.

Let's start a classifier with just that feature right and see how good it is.

Figure 2-3: Image detection through two region features [3]

This is our first feature number one and so if there's anything plausible in this region we will let it through so it is yes, if it's no then we immediately fail that region of image.so we have said for this region of image if this passes will let it through to the next stage and we will say okay it definitely could be a face.

The next feature is a three region feature and it measures the differences between the nose and the bridge and the eyes which may or may not be darker or lighter, so there's a difference there.

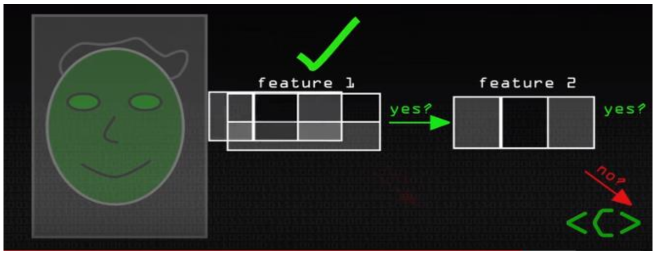


Figure 2-4: Image detection through three region features [3]

And if that passes, we go to the next feature, so this is a sort of a binary, they call it "degenerate decision there". The argument is that every time we calculate these features it a takes a little bit time. The quicker we say "no definitely not a face in there", the better. And the only time we ever need to look at all the features or all the good ones is when we think, "okay, that actually could be a face here". So, we have less and less general, more and more specific features going forward right up to about the number.

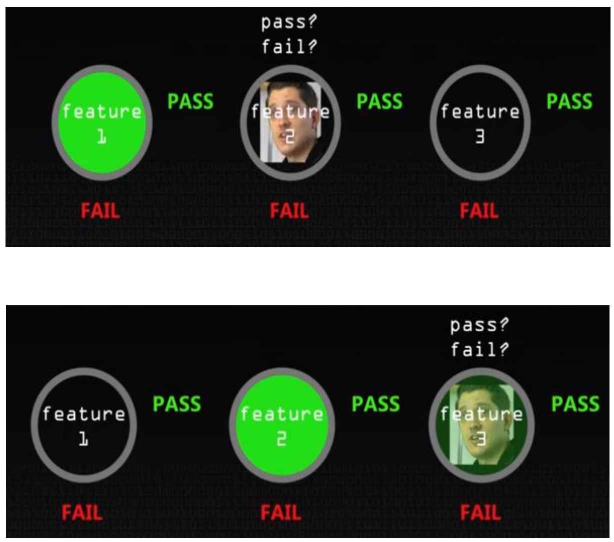


Figure 2-5: Degenerate decision process [3]

All right, so we say just the first one pass yes, just a second one pass yes, and we keep going until we get a fail and if we get all the way to the end and nothing fails that's a face, right and the beauty of this, is that for the vast majority of the image, there's no computation at all. We just take one look at it, first feature fails "not a face". They designed a really good way of adding and subtracting different regions of image and then they trained a classifier like this to find out the best features.

2.4 Drowsiness Detection

The development of a smart driving system includes drowsiness detection [4] which will be done by image processing. The base of this is detection of human face and recognition of important facial marks such as eyes. This can be done by the use of Voila Jones algorithm [2] which processes image to see where lies the eyes of humans. Further the distance is calculated between the eyes and the eye lids. Now with the information about the distance between the eyelids of open eyes comparison can be done at real time with a distance where the eyes are closed. If this condition is true for a specific period of time the driver will be declared sleeping and the alarm will sound to alert the driver [16].

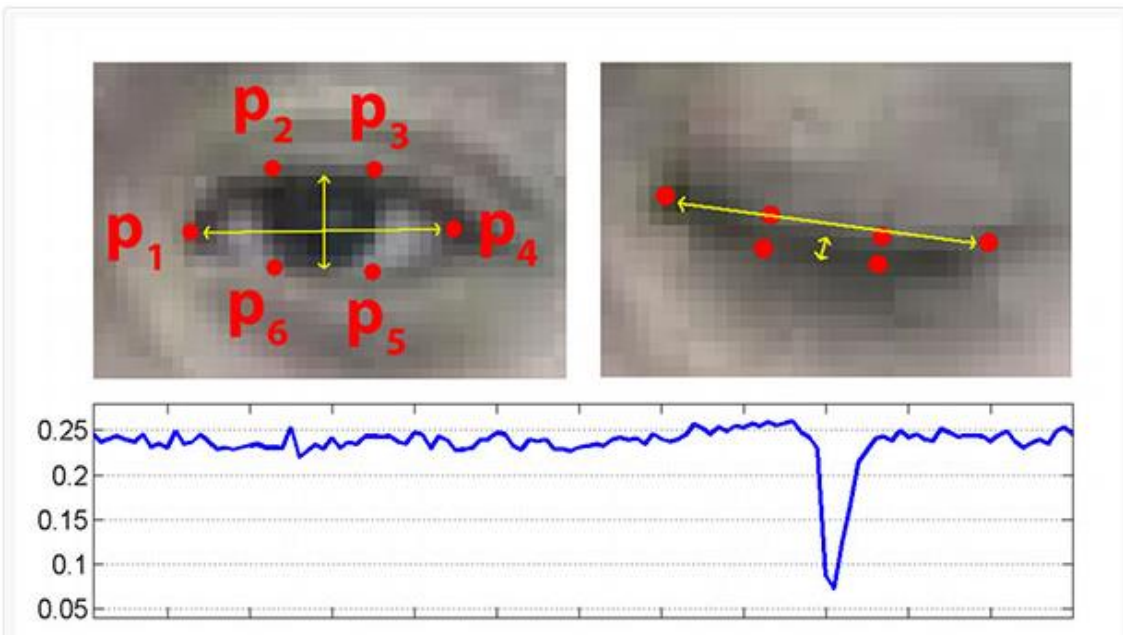


Figure 2-6: Eye aspect ratio landmarks [5]

2.4.1 Mathematical Model

Once the video is captured, we take every frame and detect important facial landmarks. Then eyes aspect ratio between the width and height of the eye is calculated.

$$\text{EAR} = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|} \qquad (1)$$

where p_1, p_2, p_3, p_4, p_5 and p_6 are landmark locations.

When the eye is open, we see that the EAR is almost same. As soon as the eye begins to close it starts getting close to zero. It is not affected by the posture or the person. This ratio is calculated by uniform scaling so it is fully invariant among individuals. Both eyes are synchronized so we take the average of EAR.

2.5 Sonar Sensor

We can measure distance between two objects by using sound ultrasonic waves, the instrument used for this process is called ultrasonic sensor [6]. Sound waves of a particular frequency are thrown towards an object. As they hit the object some are reflected back while some are diffracted and refracted. The reflected waves ae received and the distance can be calculated using the speed of sound.

Following formula is used:

$$\textbf{Distance} = \tfrac{1}{2}\,\mathbf{T} \times \mathbf{C} \qquad (2)$$

(T = Time and C = the speed of sound)

We are using sonar sensors in our project for the monitoring of the car if it’s getting close to anything on either side.

Figure 2-7: Sonar Sensor [7]

2.6 Rotary Encoder

A rotary encoder [10], is a device that is used to convert the angular motion of a shaft into analogue or digital signals. The two main types of rotary encoder are absolute and incremental.

Figure 2-8: Rotary Encoder [8]

2.6.1 Working

Following is the working principle of a rotary encoder. The square waves are generated as encoder has a disk with equally distant contact zones that are connected to the common pin C and two other contact pins A and B, as shown below.

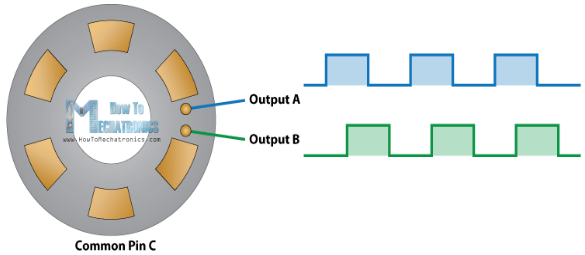


Figure 2-9: Structure of rotary encoder [9]

As soon as the disk starts moving the pins A and B will start making contact with pin C and we will have two square wave output signals. By counting the pulses of the signal, we can determine the rotated position using any two outputs. For the direction of rotation, we need to consider both signals at the same time.

As shown below the output signals are 90 degrees out of phase from each other. If the encoder is rotating clockwise the output A will be ahead of output B.

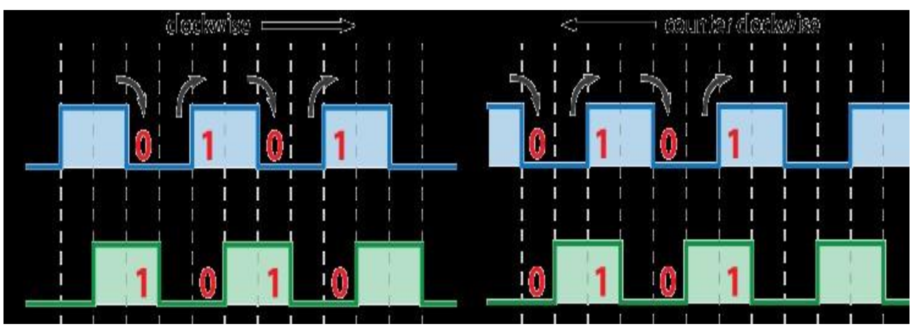


Figure 2-10: Rotational effect synchronized with clock [10]

Every step that is taken we notice that the two outputs have opposite values. This happens as the signal goes down from high to low or comes up from low to high. Clockwise rotation gives the output signal equal values. By using this we can easily program the encoder as per our requirements. We are using rotary encoders for the reading of the vehicles speed. This information will be given as input to the Arduino Due to process it.

$$\text{Speed(m/min)} = \pi * d * \text{RPM} \qquad (3)$$

Chapter 3

3 Design and Development

Design and development of Smart Driving System was done by combining two different aspects. First aspect was of digital image processing for the detection of driver's drowsiness. Second aspect included use of sensors to measure speed of the vehicle and distance of cars from either side and communicating over internet using Wi-Fi module on Node MCU.

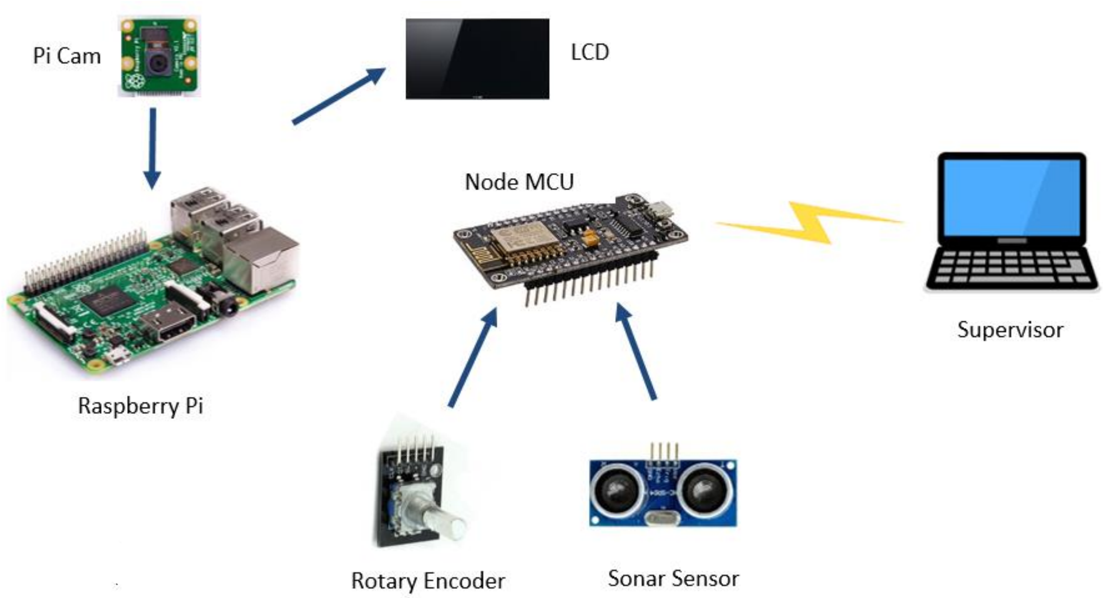


Figure 3-1: Proposed Design

3.1 Digital Image Processing

3.1.1 Development

For the process of image processing we needed to have a module that had enough processing power to process real time video imagery and applying algorithm to detect facial marks and see if the driver is awake or not. For this purpose, we selected Raspberry Pi. Raspberry Pi has processor of 1GHz and memory of 1GB. For the Raspbian software to operate on it we used an external SD card of 16 GB. Raspberry Pi comes compatible with Pi Cam which can take video imaging up to 2592 x 1944 pixels which can be brought down to any quality required fit for processing.

When the Pi Cam is on it starts taking video of the driver which is 100 MB for every minute assuming we use 5 Megapixels picture quality. Pi Cam is mounted in the vehicle right in front of the driver. This is converted into grayscale and processed real time to detect important facial landmarks. Voila Jones Algorithm [b] was used to detect the facial landmarks such as eyes. Now applying mathematical formula, we calculated eye aspect ratio that is EAR for the detection of driver's drowsiness. By using EAR we can count blinks as the EAR reaches close to zero it means that the eyes are close. Comparing the time of eyes closed with the clock, we can see how long it has been since the driver has opened the eye.

When the eye is closed for more than 10 seconds it is declared that the driver is asleep and the alert is generated.

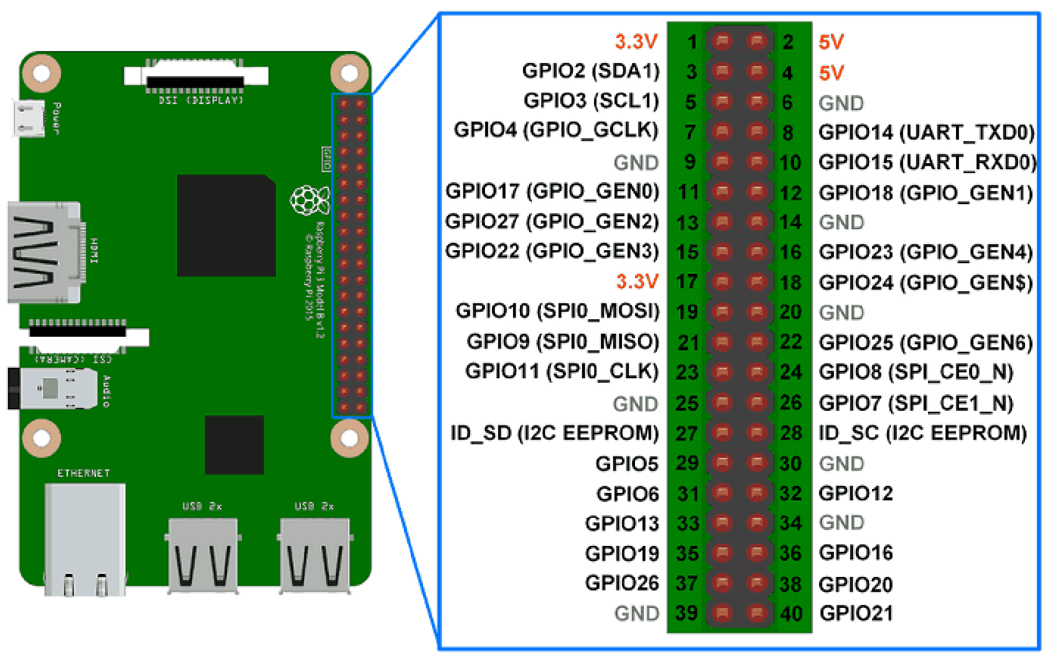


Figure 3-2: Pin Configuration of Raspberry Pi [11]

3.1.2 Hardware Design

The design included a simple module of Raspberry Pi with Pi Cam mounted on top. A 5 V micro USB power is used to turn on the module. When the module is powered up Raspbian is loaded as the boot program. For driver's drowsiness a code written on python is run. This code contains modified Voila Jones Algorithm [b] to detect eyes and measuring eyes aspect ratio which is used to determine if the eyes of the driver are closed or not.

The driver drowsiness module consists of the following components:

1. Raspberry Pi
2. Pi Cam
3. LCD
4. Mouse
5. Keyboard
6. Power source

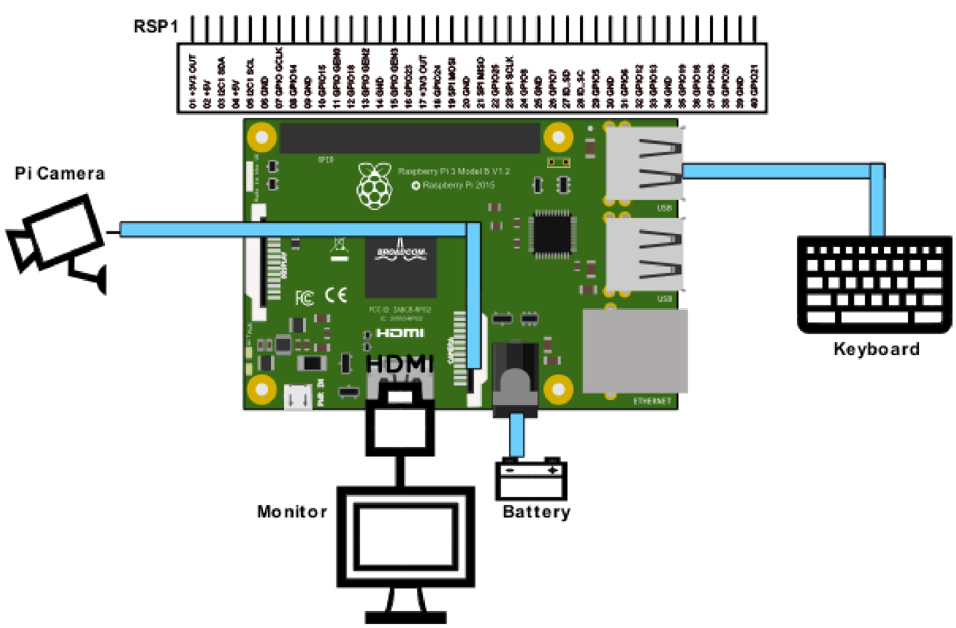


Figure 3-3: Proposed Schematic of Image Processing Module

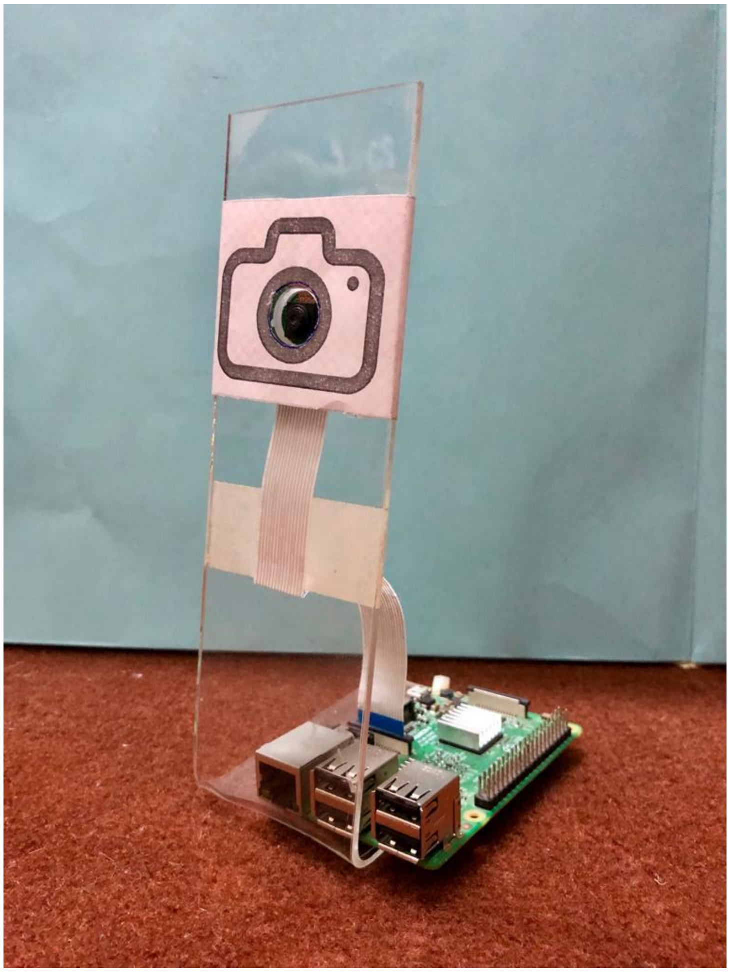


Figure 3-4: Proposed Image Processing Prototype

3.1.2.1 Eye Detection

The image once captured by the Pi Cam undergoes processing and is facial landmarks are marked. By using these markers, we are able to calculate EAR and detect if the driver is awake. Following is a picture of how the code marks eyes.

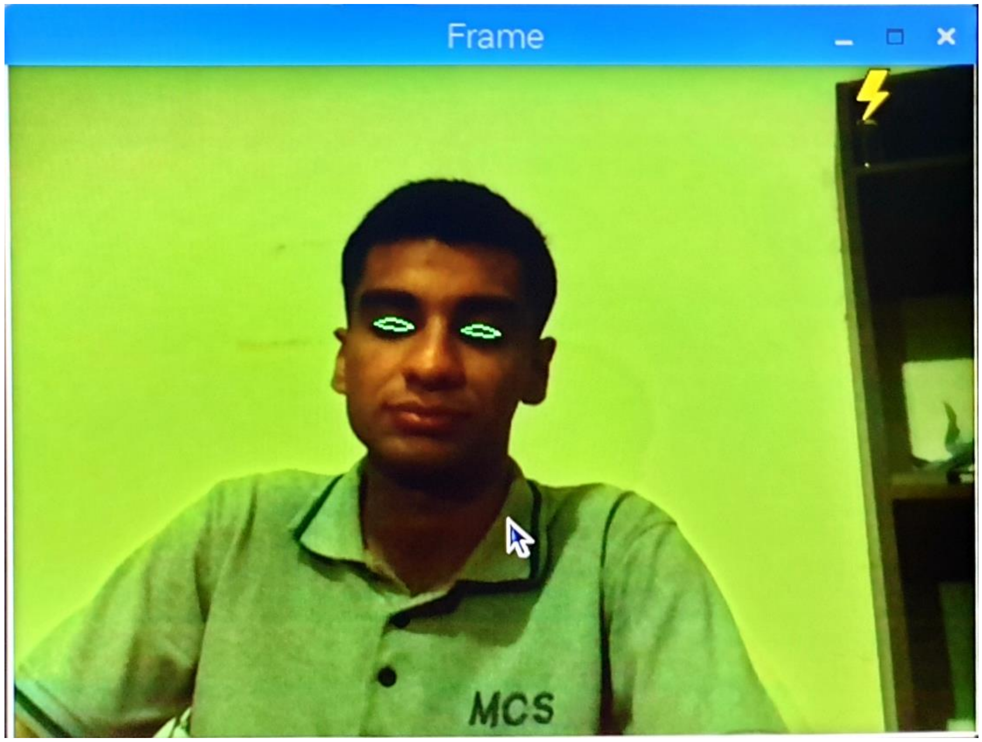


Figure 3-5: Eye detection using Pi Cam

3.1.2.2 Blinking Rate

Once the eyes are detected we use eye aspect ratio to determine state of eyes. When the eyes are closed the code syncs it with the clock to see how long is the eye closed. Normally there is a blink every second. In case the driver is asleep we have put up the threshold to be ten seconds to declare that the driver is sleeping and causing the alert.

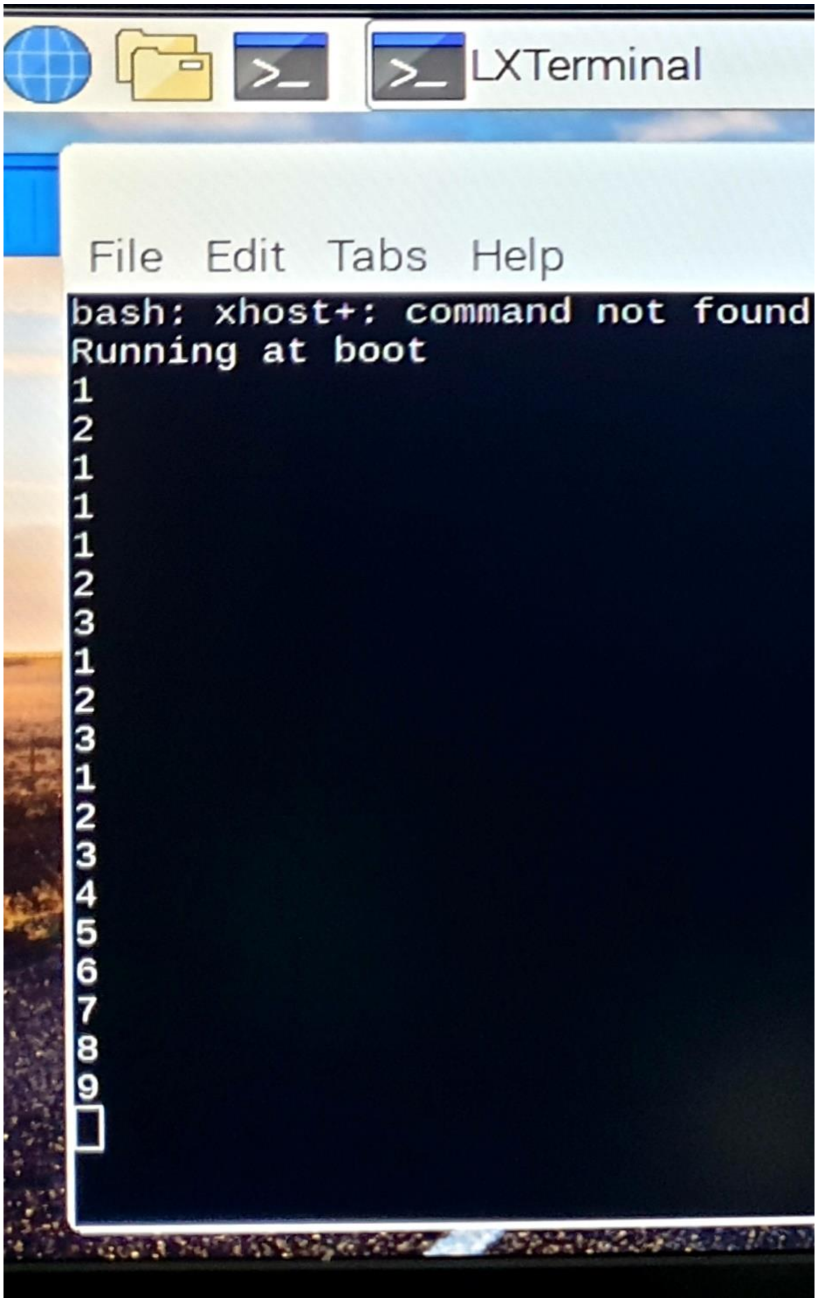
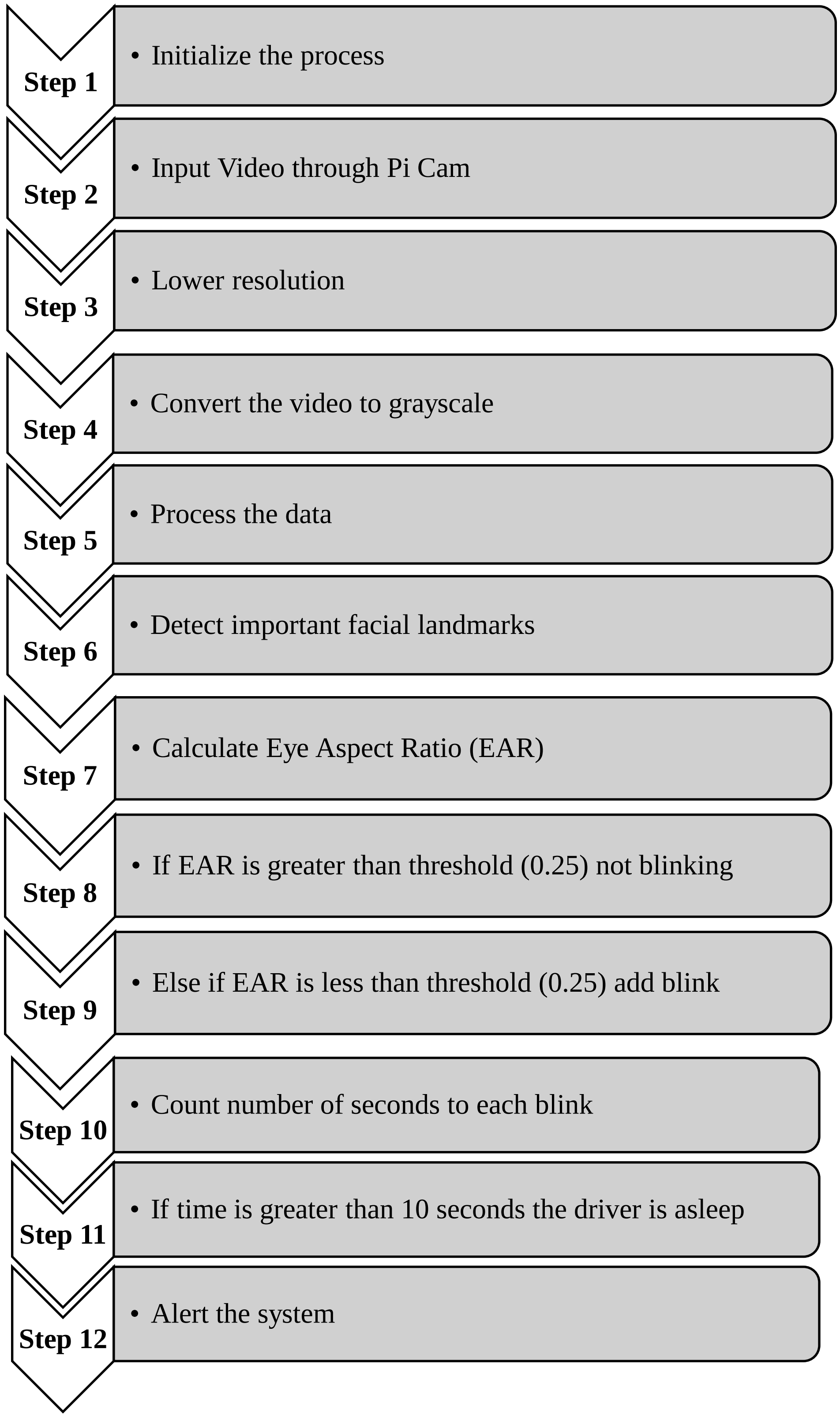


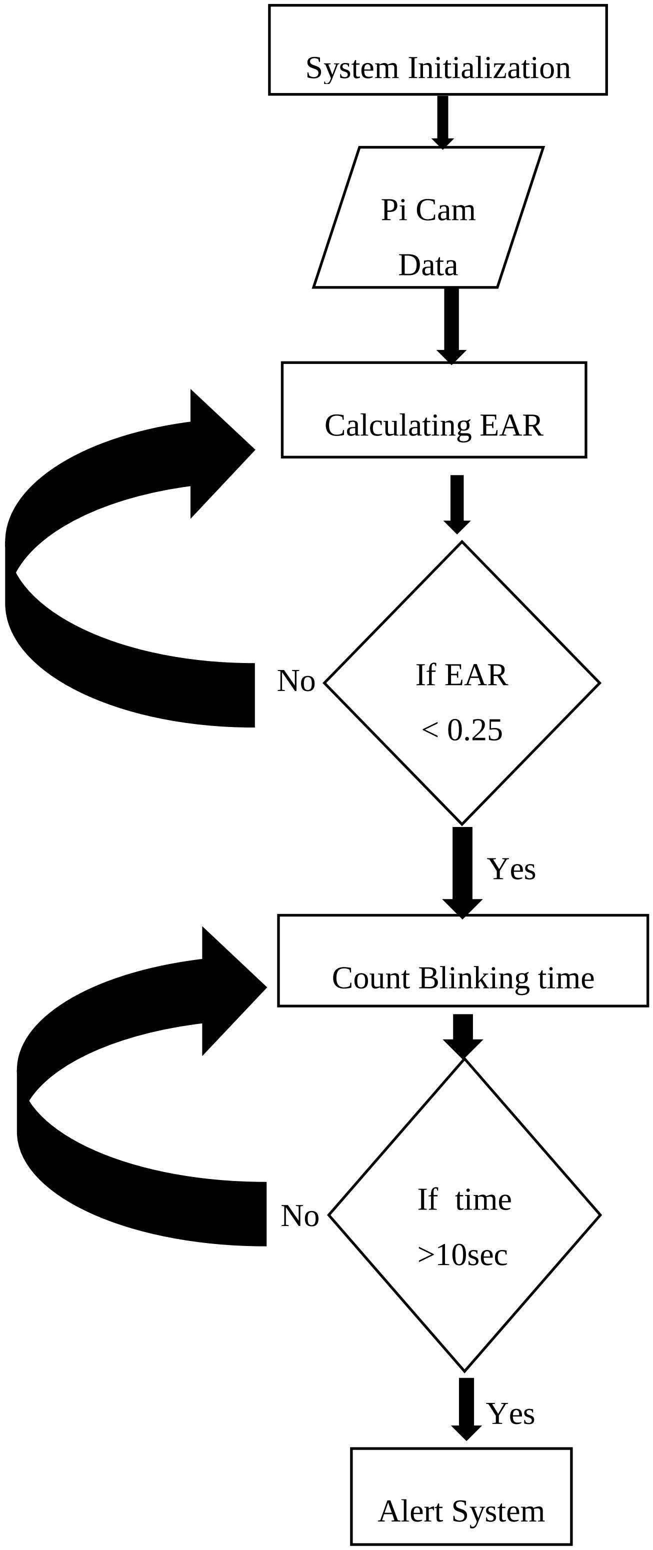
Figure 3-6: Blinks of the eye

3.1.3 Software Design

The entire process is coded in Python. The software process is explained below:


Step 1
Initialize the process
Step 2
Input Video through Pi Cam
Step 3
Lower resolution
Step 4
Convert the video to grayscale
Step 5
Process the data
Step 6
Detect important facial landmarks
Step 7
Calculate Eye Aspect Ratio (EAR)
Step 8
If EAR is greater than threshold (0.25) not blinking
Step 9
Else if EAR is less than threshold (0.25) add blink
Step 10
Count number of seconds to each blink
Step 11
If time is greater than 10 seconds the driver is asleep
Step 12
Alert the system

3.1.3.1 Flow Chart of Proposed Solution



3.2 Monitoring of vehicle speed and distance from either side

3.2.1 Development

In the second stage of our project we need to measure the speed of vehicle along with it's distance from the other objects from the either sides. For this purpose, we had two approaches, the first one was building a tachometer using the IR sensors and the other one was measuring the speed of vehicle using rotary encoders. And the distance of objects on the either sides of vehicle can be measured using Sonar sensors.

3.2.1.1 Tachometer

We had a idea of building a customizable tachometer in which we would use an IR sensor. Materials required for the customizable tachometer are

1) IR Sensor module
2) LCD Screen
3) Arduino
4) PCB
5) Header pins
6) Battery

Figure 3-7: IR Sensor [12]

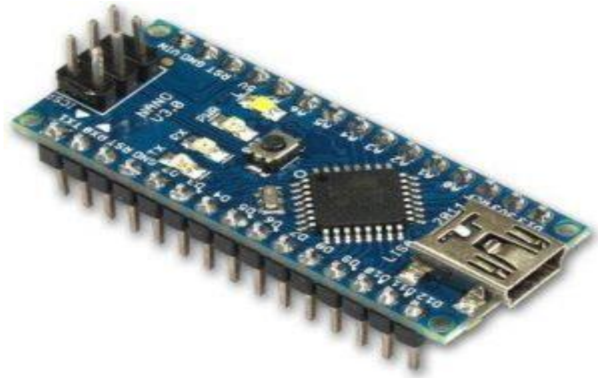


Figure 3-8: Arduino [13]

Figure 3-9: LCD [13]

The little part of rim of vehicle should have a shiny surface on the which the light of IR sensor will be thrown, every time light comes in contact with the shiny surface on the rim it counts the revolution per minute made by wheel and by using the simple formula of speed multiplying the circumference of wheel with RPM it measures the speed.

Speed = RPM x circumference of wheel (4)

During the daylight the light effect of IR sensor may not work properly as well as in the night due to lights by other vehicles this process won't be accurate moreover when the rim gets dirty on road the shiny surface will not be visible and thus tachometer will not work properly.

3.2.1.2 Rotary Encoder

Rotary Encoder is a device that can measure rotation usually is measures rotation on a shaft and translates into a series of pulses. These pulses can tell controller attached how much the shaft has moved. A rotary encoder, also called a shaft encoder, is an electro-mechanical device that converts the angular position or motion of a shaft or axle to analogue or digital output signals.

The rotary encoder will be attached with one wheel of the vehicle. As the vehicle moves rotary encoder will calculates the rotation of the wheel. The microcontroller will process the formula to measure the speed of the vehicle as programmed. The mathematical formula for calculating the speed of vehicle is

$$\mathbf{Speed(m/min) = \pi * d * RPM} \qquad (5)$$

The microcontroller we are using for our project is Node MCU.

3.2.1.3 Microcontroller Unit (Node MCU)

The Node MCU (Node Microcontroller Unit) is an open source software and hardware development environment that is built around a very inexpensive System-on-a-Chip (SoC) called the ESP8266. Node MCU is like brand name of a board that has a Wi-Fi module ESP8266 and some associated circuit. ESP8266 module in tune has a micro controller with Wi-Fi. You can program ESP8266 using Arduino.

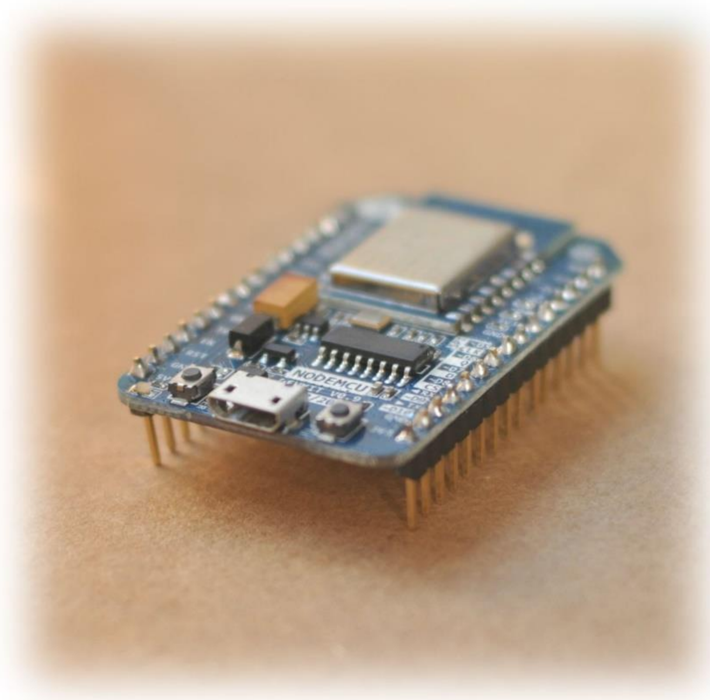


Figure 3-10: Node MCU [14]

3.2.2 Measuring the distance of vehicle from other objects

To minimize the road accidents, we have placed sonar sensors on the three sides of vehicles. The range set for the sonar sensor is set 10 cm. Anything that comes in the range of vehicles from the back and the sides will notify the driver as well as the concerned authorities through the wi-fi module on the Node MCU.

3.2.2.1 Sonar Sensor

An Ultrasonic sensor is a device that can measure the distance to an object by using sound waves. It measures distance by sending out a sound wave at a specific frequency and listening for that sound wave to bounce back. By recording the elapsed time between the sound wave being generated and the sound wave bouncing back, it is possible to calculate the distance between the sonar sensor and the object.

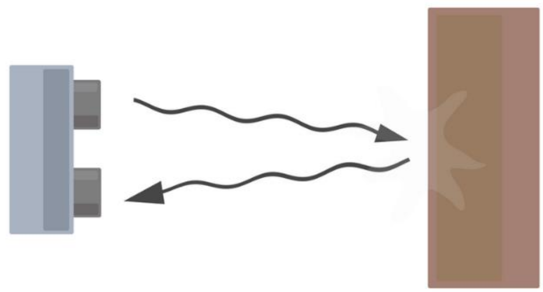


Figure 3-11: Basic ultrasonic sensor operation [15]

Since it is known that sound travels through air at about 344 m/s (1129 ft/s), you can take the time for the sound wave to return and multiply it by 344 meters (or 1129 feet) to find the total round-trip distance of the sound wave. Round-trip means that the sound wave travelled 2 times the distance to the object before it was detected by the sensor; it includes the 'trip' from the sonar sensor to the object AND the 'trip' from the object to the Ultrasonic sensor (after the sound wave bounced off the object). To find the distance to the object, simply divid the round-trip distance in half.

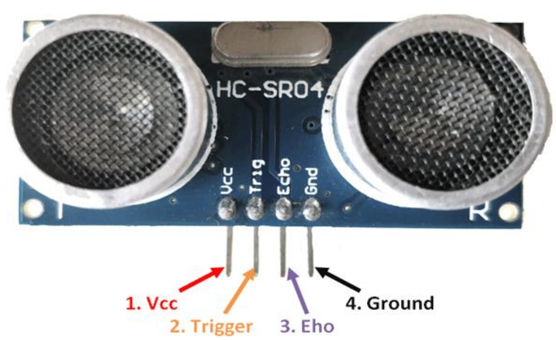


Figure 3-12: Ultrasonic sensor [7]

3.2.2.2 Sonar Sensor Pin Configuration

Table 2: Pin Configuration

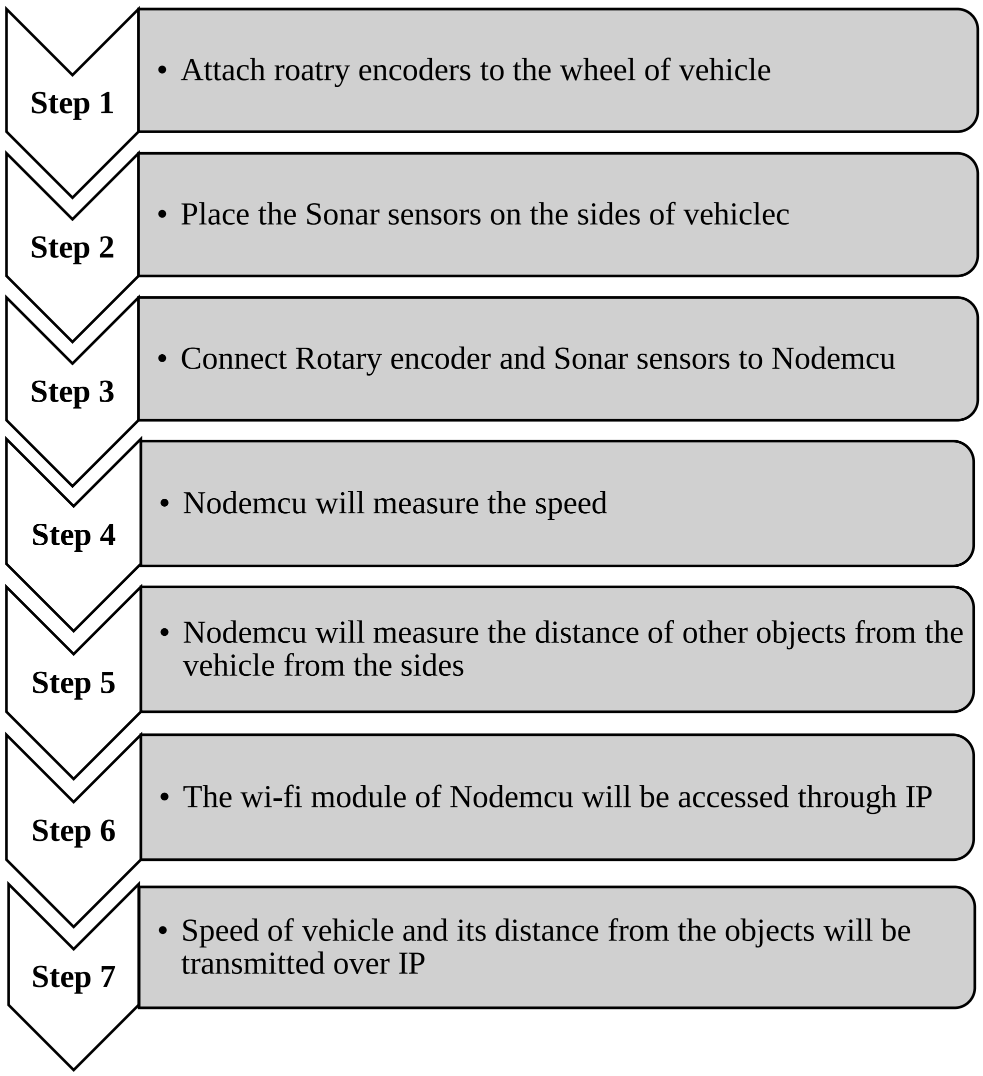
Pin Number	**Pin Name**	**Description**
1	Vcc	The Vcc pin powers the sensor, typically with +5V
2	Trigger	Trigger pin is an Input pin. This pin has to be kept high for 10us to initialize measurement by sending US wave.
3	Echo	Echo pin is an Output pin. This pin goes high for a period of time which will be equal to the time taken for the US wave to return back to the sensor.
4	Ground	This pin is connected to the Ground of the system.

3.3 Transferring data over IP

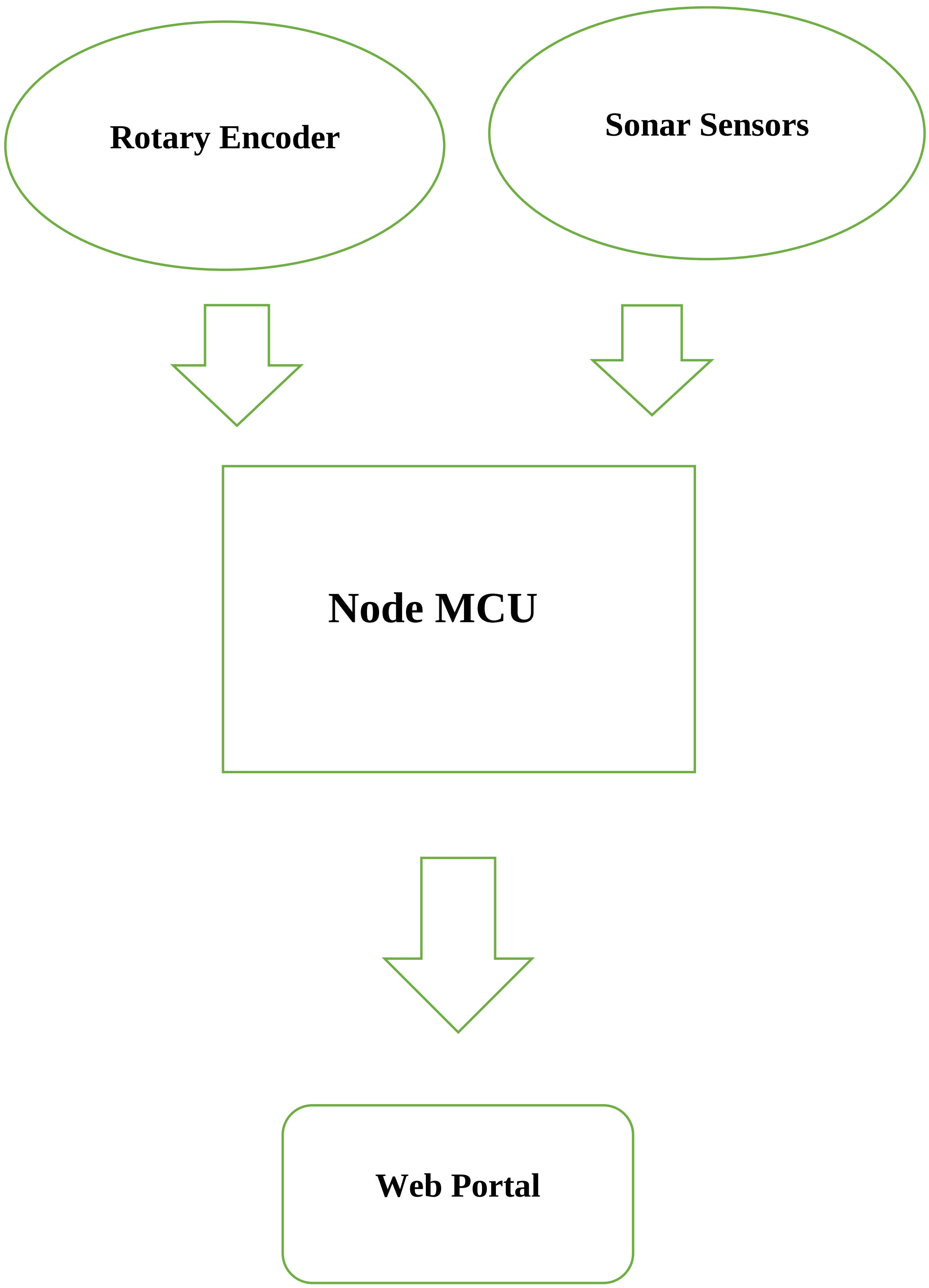
As node MCU has a wi-fi module attached over it and node MCU is connected both with the rotary encoders and sonar sensors. So, the speed of vehicle and the distance from the objects from all the sides can be checked by the concerned authorities any time over the IP. For this purpose, we have developed a web portal and anybody having the IP address of the wi-fi module can have access to speed of vehicle.

3.3.1 Software Design

The software process is explained below:



3.3.1.1 Flow Chart of Proposed Solution



Chapter 4

4 Analysis and Evaluation

We have analysed our project in terms of the following:

1. Design
2. User Friendly

4.1 Design

Smart Driving System has following advantages:

Table 3: Advantages of design

Sr	**Advantages**
1.	Cost effective
2.	Easy to install
3.	Very low maintenance

Following are its disadvantages

Table 4: Disadvantages of design

Sr.	**Disadvantages**
1.	Processing delay
2.	Fragile

4.2 User Friendly

The smart driving system is as user friendly and compatible as it gets. The driver drowsiness detection part is a Pi Cam mounted on Raspberry Pi module which becomes a little module weighing less than 200 grams. It can easily be mounted anywhere in front of the driver that is either on the dashboard or the roof top the only requirement being it should be able to cover the driver's seat.

This module in no way hinders the driver's view of the road which its previous prototypes were failed in doing. IR sensor-based glasses for drowsiness detection blinded the driver with one eye. Which caused a major discomfort in driving.

Moreover, the prototype suggested measuring speed using customizable tachometer. This can be any tachometer compatible with the vehicle, as the module only needs vehicle speed input. Once this input is extracted it is easily transferred via internet making the data of real time movement of vehicle readily available.

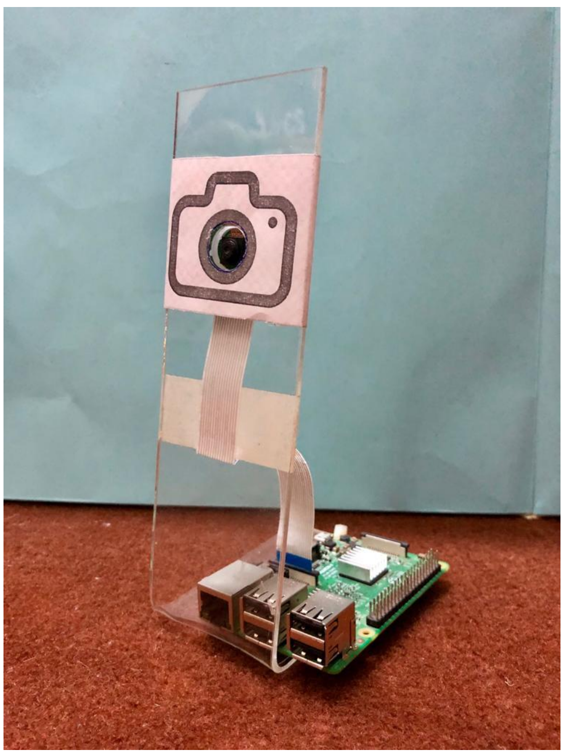


Figure 4-1: User Friendly prototype

Chapter 5

5 Conclusion

As our project is basically for the military use but this smart driving system is applicable in both military as well as in the private sector. In military it is very helpful in saving the military asset and manpower. We know as the world is progressing the traffic on the roads is increasing day by day thus the danger to road accidents is increasing with the same ratio. This project is an effort and will be helpful in the future in this regard as well. The information about the eyes of person is obtained through the algorithm of image processing. When the eyes are kept closed for a specific period of time, an alarm is generated. In addition, during monitoring, the system is able to automatically detect any eye localizing error that might have occurred. In case of this type of error, the system is able to recover and properly localize the eyes. The following conclusions were made:

1. Image processing achieves highly accurate and reliable detection of drowsiness.

2. Image processing offers invasive approach to detecting drowsiness without the annoyance and interference.

3. A drowsiness detection system developed around the principle of image.

4. Image processing judges the driver's alertness level on the basis of continuous eye closures.

5. The speed measuring can generate the alarm when the driver over speeds.

The transmission of information over the web portal makes it easy for the concerned authorities to keep a check.

Chapter 6

6 Future Work

A prototype of smart driving system has been designed. The improvements that can be made to the system are listed below:

1. The distance covered by the vehicle can also be calculated as we have calculated the speed by measuring the time as vehicle uses the system.

$$\text{Distance} = \text{speed} \times \text{time}$$

2. A GPS module can be attached with nodemcu that can determine the location of vehicle at any time and transmit the information to the concerned authorities at any time.

3. The delay in reading the eyes and in generating the alarms when eyes are closed can be minimized for a more effective system.

4. The system can be made more efficient in the darkness as well by use of some good quality camera.

5. The project can be highly applicable to automobile industries and will be more efficient when it will come in a vehicle, fitted by the industry.

6. Transport companies (e.g. Daewoo bus service) can use this smart driving system in their vehicles to keep a check on the vehicles remotely.

References

1. https://en.wikipedia.org/wiki/Digital_image_processing

2. P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, Kauai, HI, USA, 2001, pp. I-I.

3. https://www.youtube.com/watch?v=uEJ71VlUmMQ&t=4s

4. A. Malla, P. Davidson, P. Bones, R. Green and R. Jones, "Automated Video-based Measurement of Eye Closure for Detecting Behavioral Microsleep", in 32nd Annual International Conference of the IEEE, Buenos Aires, Argentina, 2010.

5. https://images.app.goo.gl/N1p4PUmcaHvrvx5QA

6. https://www.keyence.com/ss/products/sensor/sensorbasics/ultrasonic/info/index.jsp

7. https://images.app.goo.gl/HX9Z6cms2f7kM7q88

8. https://images.app.goo.gl/RTWYog5oCEXXHJDe7

9. https://images.app.goo.gl/Dn5naBSMGPsr2Bmh9

10. https://howtomechatronics.com/tutorials/arduino/rotary-encoder-works-use-arduino/

11. https://images.app.goo.gl/eAqNKdGkXyEfPmGS9

12. https://images.app.goo.gl/i5oY9PqJ3GiENoWB6

13. https://www.how2electronics.com/digital-tachometer-using-ir-sensor-with-arduino/

14. https://www.nodemcu.com/index_en.html

15. http://cmra.rec.ri.cmu.edu/content/electronics/boe/ultrasonic_sensor/1.html

16. S. Vitabile, A. Paola and F. Sorbello, "Bright Pupil Detection in an Embedded, Real-time Drowsiness Monitoring System", in 24th IEEE International Conference on Advanced Information Networking and Applications, 2010.

APPENDIX A

Image Processing Code:

```
from scipy.spatial import distance

from imutils import face_utils

import imutils

import dlib

import io

import picamera

import numpy

import cv2

import numpy as np

import time

import RPi.GPIO as GPIO



stream = io.BytesIO()



Pin1 = 33
```

```
GPIO.setmode(GPIO.BOARD) # Numbers GPIOs by physical location

GPIO.setwarnings(False)

GPIO.setup(Pin1, GPIO.OUT)

GPIO.output(Pin1, GPIO.LOW)



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



#Get the picture (low resolution, so it should be quite fast)

#Here you can also specify other parameters (e.g.:rotate the image

with picamera.PiCamera() as camera:

    camera.resolution = (640, 480)

    camera.start_recording(stream, format='h264', quality=23)

    camera.wait_recording(15)
```

```
    camera.stop_recording()


#Convert the picture into a numpy array

buff = numpy.fromstring(stream.getvalue(), dtype=numpy.uint8)



thresh = 0.25

frame_check = 10

detect = dlib.get_frontal_face_detector()

predict = 
dlib.shape_predictor("/home/pi/Drowsiness_Detection/shape_predictor_6
8_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('video.h264')



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)

					GPIO.output(Pin1, GPIO.HIGH)

					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.stop()
```

APPENDIX B

Node MCU Code:

```
#include <Encoder.h>

#include <wifiserver.h>



int trig1 = 16,trig2 = 4,trig3 = 14;

int echo1 = 5,echo2 = 2,echo3 = 12;

int distance1,distance2,distance3;

double rpm, speeD;

//int buzzer = 1, drowsy = 3;

int safe = 10 , limit = 50;

long oldPosition  = -999;



wifiserver wlan("ADEEL BROTHERS", "ATmega328");

Encoder myEnc(15, 13);



void sonar_init(int trigpin, int echopin){

pinMode(echopin, INPUT);
```

```
pinMode(trigpin, OUTPUT);
}


int check_distance(int trig_pin, int echo_pin){
digitalWrite(trig_pin,LOW);
delayMicroseconds(2);
digitalWrite(trig_pin,HIGH);
delayMicroseconds(10);
digitalWrite(trig_pin,LOW);
int duration = pulseIn(echo_pin,HIGH);
return (duration*0.034/2);
}


void setup() {
  server_init();
  sonar_init(trig1,echo1);
  sonar_init(trig2,echo2);
  sonar_init(trig3,echo3);
```

```
//  pinMode(drowsy,INPUT);

  // put your setup code here, to run once:



}



void loop() {



distance1 = check_distance(trig1,echo1);

distance2 = check_distance(trig2,echo2);

distance3 = check_distance(trig3,echo3);

long newPosition = myEnc.read();

rpm = (double)(newPosition - oldPosition);

oldPosition = newPosition;

speeD = rpm * 20.41;

speeD/= 60;

send_string("Left distance: " + (String)distance1 + ", Center Distance: " +
(String)distance2 + ", Right Distance: " + (String)distance3 + ", Speed: "
+ (String)speeD + "m/s" + " RPM:"+ (String)  rpm);

//  // put your main code here, to run repeatedly:
```

```
//  tone(tonePin,200000,200);

//  tone(tonePin,5000,200);

//  tone(tonePin,1000,200);

if(distance1 <= safe || distance2 <= safe || distance3 <= safe){

  //tone(buzzer,2000,200);

}

else if(rpm >= limit){

  //tone(buzzer,500,200);

}

//else if(digitalRead(drowsy) == HIGH){

//  tone(buzzer,1000,200);

//}

else{

  //noTone(buzzer);

}

}
```