VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, Belagavi-590018, Karnataka

Project Report

On

"AUTOMATED DRIVING MANAGEMENT WITH IGNITION CONTROL"

Submitted in partial fulfillment of the requirement for the award of the degree of

Bachelor of Engineering

In

Electronics and Communication Engineering

Submitted by

LAKSHMIKANTH L	**1VI19EC070**
LOHITH H P	**1VI19EC073**
NIKHIL G N	**1VI19EC094**
PATANI VED SHARADKUMAR	**1VI19EC101**

Under the guidance of

Dr. HEMAMALINI G E

Associate Professor

Department of Electronics & Communication Engineering

VEMANA INSTITUTE OF TECHNOLOGY

(Affiliated to VTU & Approved by AICTE)

#1, Mahayogi Vemana Road, Koramangala, Bengaluru– 560034

2022-23

ABSTRACT

Drunk and Drowsy driving is dangerous. This usually happens when a driver has not slept enough, or due to untreated sleep disorders medications, drinking alcohol, or shift work. The objective of this work is to develop a system for driver drowsiness detection to prevent accidents, in addition to this system proposes the alcohol detection of the driver with ignition control. The system incorporates the Raspberry Pi, MQ-3 alcohol sensor and L298N motor driver. It uses image processing technique called (CNN) Cascaded Convolutional Network architecture which uses region of interest (ROI) to extract the facial feature points and detect the states of the eyes, pupil and mouth over time. The pupil, iris and mouth area are calculated and then it compared with threshold value, If it exceeds the threshold value then driver drowsiness (eye and yawning) condition is detected, alcohol level is detected with the ignition control and alarm indicated by a taillamp and buzzer in order to wake the driver immediately. The system sends an alert SMS to the person whose contact is saved in the system using TWILIO API (Application Program Interface), SMS will be sent to registered mobile number.

ACKNOWLEDGEMENT

We sincerely thank **Visvesvaraya Technology University (VTU)** for providing a platform to do the project work.

We express our sincere thanks to **Dr. Vijayasimha Reddy B G**, Principal, **Vemana Institute of Technology**, Bengaluru, for providing necessary facilities and motivation to carry out project work successfully.

We express heartfelt gratitude and humble thanks to **Dr. Chandra Shekar S M**, HoD, ECE, **Vemana Institute of Technology**, for his constant encouragement, inspiration and help to carry out project work successfully.

We would like to express our sincere gratitude towards our internal guide **Dr. Hemamalini G E**, Associate Professor for providing encouragement and inspiration for this work.

We thank project coordinators **Dr. Harshada J Patil** and **Prof. Prathima A** for their cooperation and support during the project work.

We are thankful to **all the teaching and non-teaching staff members** of Electronics & Communication Engineering department for their help and needed support rendered throughout the work.

LAKSHMIKANTH L **(1VI19EC070)**

LOHITH H P **(1VI19EC073)**

NIKHIL G N **(1VI19EC094)**

PATANI VED SHARADKUMAR (1VI19EC101)

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Drowsy driving is the dangerous combination of driving and sleepiness or fatigue. This usually happens when a driver has not slept enough, or due to untreated sleep disorders, medications, drinking alcohol, or shift work. The National Highway Traffic Safety Administration (NHTSA) and World Health The taxi drivers, bus drivers, truck drivers and people traveling long-distance suffer from lack of sleep, which is dangerous and causes mishaps on the roads. Most accidents happen due to the drowsiness of the driver, caused by tiredness, tedious road condition and unfavorable climate situations. Organization (WHO) have reported that approximately 1.35 million people die every year due to vehicle crashes acrossthe world. These situations arise if the driver is addicted to alcohol or in drowsiness. So, toprevent these accidents it is necessary to build a system using the latest techniques such asa real time intelligent driver assistant system and alcohol monitoring system in the vehicles.

The instances of traffic accidents because of drunk driving have increased quickly. More and more people have realized that drunk driving has ruptured public safety on roads. It's time to develop a kind of system which can stop the drunk driving effectively. As this kind of system hasn't been popularized, we try to develop this system to be available on every car. This system won't cost much, but it will bring much for its effectiveness on human's safety. This system has a preventative effect which can stop accidents from the beginning. The utilization of liquor, even in little amount, raises the risk of being engaged in an accident for drivers and people on foot. In addition to the fact alcohol decreases the ability to processes which are important for road safety, such as visibility and reaction time, it is also associated with impaired judgement, speeding or not using seat-belts. In other nations, research shows that ample number of drivers, bike riders and people on foot have liquor in their blood in adequate concentrations to hamper their road use abilities For instance, youth drivers are at a high danger of such crashes, and crashes involving alcohol are more incessant after evening. Sadly, in many nations, the extent of the problem is not well understood, there is minimal public consciousness of the issue and legislation and authorization are insufficient. The taxi drivers, bus drivers, truck drivers and people traveling long-distance suffer from lack of sleep, which is dangerous and causes mishaps on the roads. Most accidents happen due to the drowsiness of the driver.

1.1 Aim

To determine the driver drowsiness and alcohol detection with Ignition Control.

1.2 Objectives

- The main objective of this product is to automatically detect driver drowsiness levelalso detects the alcohol level of driver and warns him/her regarding their state in order to avoid any mishap caused by them and sends an alert SMS to the person whose contact is saved in the system using TWILLO API (Application Program Interface) SMS will be sent to registered mobile number.
- This can be made as a cost-effective product for the Indian truck drivers, public service vehicles and private vehicle drivers across India.
- Automation can help reduce the number of crashes on our roads.

CHAPTER 2

LITERATURE SURVEY

[1] "Drunk and Drive Detection using Raspberry-Pi" A. Siri Pallavi1, K. Varun Kumar, T. Vamsi Krishna, S. Sandeep, T. S. Jyothi Lakshmi".

This paper presents the system design and concept of an alcohol detection. It is a System on drunk and drive detection using Raspberry-PI. The system uses raspberry pi with alcohol sensor and dc motor, achieve this purpose. System uses alcohol sensor, raspberry pi with dc motor to demonstrate as vehicle engine. System constantly monitors the sensitivity of alcohol sensor for drunk driver detection. If driver is drunk, the processor instantly stops the system ignition by stopping the motor. If alcohol sensor is not giving high alcohol intensity signals, system lets engine run. At the same time, it is connected to a network from where the person who is driving is being monitored. Here MQ-3 sensor has in built potentiometer where it works as threshold level for the sensor. MQ-3 sensor converts the PPM value to analog value and operates by comparator logic. The potentiometer value is connected to second pin of the comparator where it acts as reference value for the sensor. When the PPM value crosses the reference, voltage set by potentiometer then comparator gets output as 1 which shows alcohol is sensed by the sensor or else the output is zero where no alcohol is detected by the sensor.

[2] "Design and Implementation of Driver Drowsiness and Alcohol Intoxication Detection Using Raspberry-PI" Pandurang N. Kathar,Prof. D.L.Bhuyar.

This paper presents the system design and concept of an alcohol detection. It is a System to design and implementation of driver drowsiness and alcohol intoxication detection using Raspberry-PI. Besides eye and head movements, another visual cue that can potentially capture one's level of drowsiness is his/her eyes and faces detection analysis. Making a real time application with computer vision is very effective and efficient challenging task that needs processing powerful system. OpenCV is opensource software, which is used for creating computer vision. OpenCV is available in C, C++, and Python and Java programming languages extension. Raspberry controller small sized ARM 11 opensource controller with the GPU provides up to 1.5Gpixels of graphics processing700 MHz. It can be over clocked maximum 1500MHz Raspberry-pi can work with Raspbian operating system, which is a lightweight Linux. Raspbian OS is loaded withprogramming software

and OpenCV. It supports interfacing of various low level and high- level peripherals including USB camera and GPIO's. In case of driver is in sleepy or finding fatigue, the message will be sent by using GSM and buzzer will be turned on till theGSM positive message from car owner. . Haar Feature based Cascade Classifier technique,it is a machine learning based approach where a cascade function is trained from a lot ofpositive and negative images, and this positive image is used for detecting face region andeye region the update of region of interest ROI. Open CV is packed with a trainer as wellas detector. The open CV is used for creating user defined object classifier. The objectclassifier that has been created is stored in.xml file extension classifier can be used in thelater stages of programming, also in this paper we use canny operator edge detection forrecognize exact coordinate of eyes region. On the other hand of the system arduino is usedfor detection of the alcohol consumption by the person, alcohol gas sensor or breathalyzer MQ-3 is interfaced. Arduino will detect samples of the person who is driving drunk or not.Based on the output from Arduino, an alarm will be turned on and the car's ignition powersource can be cut down through a relay to stop the car or preventing the driver to start thecar. If driver is in over limit drunken, then the message will be sent by using GSM andbuzzer will be turned on till the GSM positive message from car owner

[3] "Drunk and Drive Detection with Ignition Lock Using Raspberry PI"Anil Kumar Patnik , Mondi KVS Manichandu, J.S.S. Keerthi, G. Satish, Y. Chanukya.

This paper presents the system design and concept of an alcohol detection. It is a Systemon Drunk and Drive detection with ignition lock using Raspberry-PI. The Hardwaremainly consists of Raspberry pi3, Alcohol Sensor MQ3, L293D motor driver board,Twilio, LCD display, Buzzer, Ignition lock. Initially for Raspberry pi Power supply isgiven through USB cable (or) by Mobile charger. First Raspbian OS is installed inraspberry pi, then the components are connected and they have to write the code for different hardware components. After executing the codes, the circuit works. Initially the engine gets started immediately when drunken driver enters into vehicle. Sensor measures the alcohol content if he exceeds the limit then Twilio sends SMS to driver and higher authorities and the SMS is displayed on the screen i.e. LCD and gives buzzer. Thus the vehicle will be stopped on detecting alcohol concentration. When the PPM value crosses the reference, voltage set by potentiometer then comparator gets output as 1 which shows alcohol is sensed by the sensor or else the output is zero whereno alcohol is detected by the sensor.

[4] "Drowsiness Detection Using RASPBERRY-PI Model Based on Image Processing" Miss.Rajeshwari Sanjay Rawal ,Mr. Sameer. S. Nagtilak.

This paper presents the system design and concept of a drowsiness detection. It is a raspberry-pi model based on image processing system design implemented to check out and detect driver drowsiness and also preventing the road accidents. It contains camera, Raspberry-PI, Serial Interfacing, PIC Controller, Buzzer and Relay Switch. This proposed system consists of open source 5-megapixel digital camera for capturing real time images of car driver. For further processing on that image, we need to send the imageto Raspberry-pi system board. The Raspberry-pi system is loaded with Raspbian OS andPython packages for Open CV (Computer Vision). Haar features are used to calculate required part of the eye (pupil and iris). Further, Hough transform is used for edge detection of pupil and iris. Pupil and Iris area is calculated and then it compares with threshold value. If it exceeds the threshold value then driver drowsiness condition is detected and alarm indicated by buzzer. Buzzer is directly connected to PIC 16F controller and the PIC controller is serially interfaced with Raspberry-pi system board. The idea of this project is to only detect driver drowsiness. Haar cascade Face region, Eye region and Open eye region is calculated.

[5] "IOT Based Driver Drowsiness Detection and Safety Monitoring System Using Raspberry-PI" Jasmina D Vora, Sharmila A, Yoganandhini S, Aparna R.

This paper focuses on crop protection system which is a major threat in India. In this the author has used has used Raspberry pi along with the sensors and camera are interfaced to it. Agriculture plays the major role in economics and survival of people in India. IoT is widely used for connecting device and collecting data information. The main aim is to prevent the loss of crops and to protect the area from intruders and wild animals which pose a major threat to the agricultural areas. Such a system will be helpful to the farmers in protecting their orchards and fields and save them from significant financial losses and saves them from unproductive efforts that they endure for the protection of their fields. When the PIR sensor detects the motion camera will be turned on and capture the image. The captured Image is stored in Board as well as in the cloud. Simultaneously a message will be generated and send information about intrusions to owner using a SIM900A module along with the details of the temperature and humidity. If the motion detection is due to an authorized person with a valid RFID, if he is a farm worker, then his attendance gets

recorded automatically. This project carriers a great social relevance as it aims to address the problem. Thus, a smart embedded farmland protection surveillance-based system which is low cost, and consumes low energy.

[6] "Real-Time Driver Drowsiness Detection System Using Eye Aspect Ratio and Eye Closure Ratio" Sukrit Mehta, Sharad Dadhich, Sahil Gumber, Arpita Jadhav Bhat.

This paper presents the system design and concept of a drowsiness detection. It is an Real time based system design implemented to check out and detect driver drowsiness using eye aspect ratio and eye closure ratio. This section details the proposed approach to detect driver's drowsiness that works on two levels. The application is installed on driver's device running Android operating system (OS). The process starts with capturingof live images from camera and is subsequently sent at local server. At the server's side, DLib library is employed to detect facial landmarks and a threshold value is used to detectwhether driver is drowsy or not. These facial landmarks are then used to compute the EAR (Eye Aspect Ratio) and are returned back to the driver. In our context, the EAR value received at the application's end would be compared with the threshold value takenas 0.25.If the EAR value is less than the threshold value, then this would indicate a stateof fatigue. In case of Drowsiness, the driver and the passengers would be alerted by an alarm.

[7] "A System on Intelligent Driver Drowsiness Detection Method" Ugra Mohan Kumar, Devendra Singh , Sudhir Jugran , Pankaj Punia , Vinay Negi.

This paper presents the system design and concept of a drowsiness detection. It is a System on Intelligent Driver Drowsiness Detection Method. This drowsiness detection is based on Artificial Neural Network learning and the data fusion of a few advanced pointers acquired. Driver bodily pointers get information from vision system. The camera touches to the NIR, are set before the driver to obtain pictures of the driver freely of encompassing striking surroundings. In case of bad illumination, the NIR consequently remain on and the other way around. Picture handling method to handle the eye detection/tracking and conclusion parametrization to estimate pieces of information.

CHAPTER 3

DESIGN METHODOLOGY

Design methods are procedures, techniques, aids, or tools for designing. They offer a number of different kinds of activities that a designer might use within an overall design process. Design methodology is the broader study of method in design: the study of the principles, practices, and procedures of designing. The key to design methodology is finding the best solution for each design situation, whether it be in industrial design, architecture, or technology. Design methodology stresses the use of brainstorming to encourage innovative ideas and collaborative thinking to work through each proposed idea and arrive at the best solution. Meeting the needs and wants of the end user is the most critical concern. Design methodology also employs basic research methods, such as analysis and testing.

3.1 System Design

The explanation behind the plan is to orchestrate the course of action of the issue dictated by the necessities report. This stage is the underlying stage in moving from issue to the game plan space. All things considered, start with what is obliged; diagram takes us to work towards how to satisfy those necessities. The design of the system is perhaps the most essential segment affecting the way of the item and note worthily affects the later stages, particularly testing and upkeep. System diagram delineates all the huge datastructure, report game plan, yield and genuine modules in the system and their Specification is picked

3.2 System Architecture

The architectural configuration procedure is concerned with building up a fundamental basic system for a framework. It includes recognizing the real parts of the framework and interchanges between these segments. The beginning configuration procedure of recognizing these subsystems and building up a structure for subsystem control and correspondence is called construction modelling outline and the yield of this outline procedure is a portrayal of the product structural planning. All things considered, start with what is obliged; diagram takes us to work towards how to satisfy those necessities. The proposed architecture for this system is given below. The design of the system is perhaps the most essential segment affecting the way of the item and note worthily affects the later stages, particularly testing and upkeep. It shows the way this system is designed and brief working of the system as illustrated in below Fig 3.1 below,

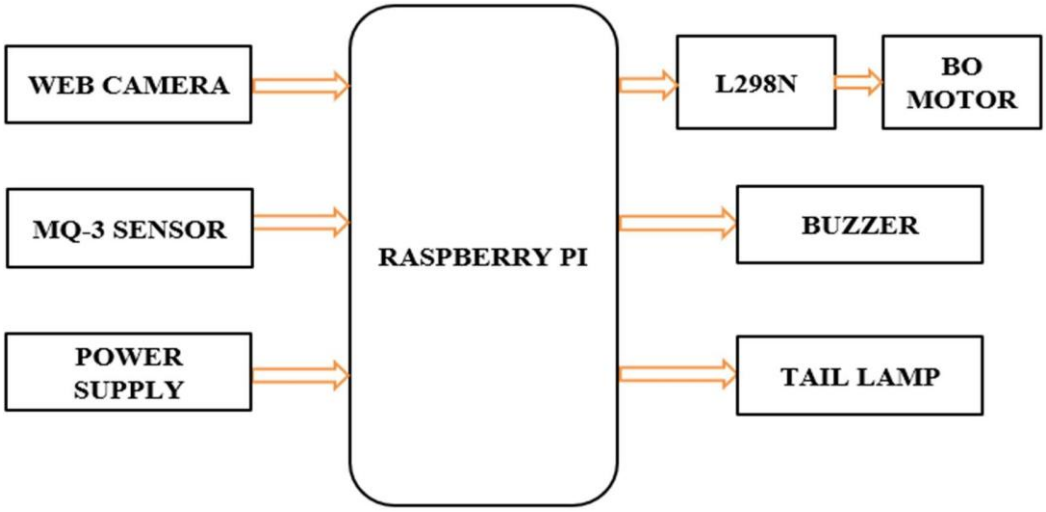

Fig 3.1 : System Architecture

3.3 Hardware Requirements

- Raspberry Pi 3 Model B
- Web Camera
- L298N Driver
- MQ-3 Sensor
- Buzzer
- Bo Motor
- 220ohm resistor
- Bread Board
- Tail Lamp

3.3.1 RASPBERRY PI 3 B

- RASPBERRY PI 3 B Raspberry PI is a series of small single board computers (SBCs) developed in the UNITED KINGDOM by the RASPBERRY PI FOUNDATION in association with Broadcom the Raspberry Pi project originally leaned towards the promotion of teaching basic computer science in schools and in developing countries The original model became more popular than anticipated, selling outside its target market for uses such as robotics. It is widely used in many areas, such as for weather monitoring because of its low cost, modularity, and open design. It is typically used by computer and electronic hobbyists, due to

its adoption of the HDMI and USB standards.

- Raspberry Pi 3 Model B was released in February 2016 with a 1.2 GHz 64-bit quad core ARM Cortex-A53 processor, on-board 802.11n Wi-Fi, Bluetooth and USB boot capabilities
- On Pi Day 2018, the Raspberry Pi 3 Model B was launched with a faster 1.4 GHz processor, a three-times faster gigabit internet (throughput limited to ca. 300 Mbits/s by the internal USB 2.0 connection), and 2.4 / 5 GHz dual band 80.11ac Wi-Fi (100 Mbit/s).
- Other features are Power over Ethernet (PoE) (with the add-on PoE HAT), USB boot and network boot (an SD card is no longer required) as shown in Fig 3.1 below.

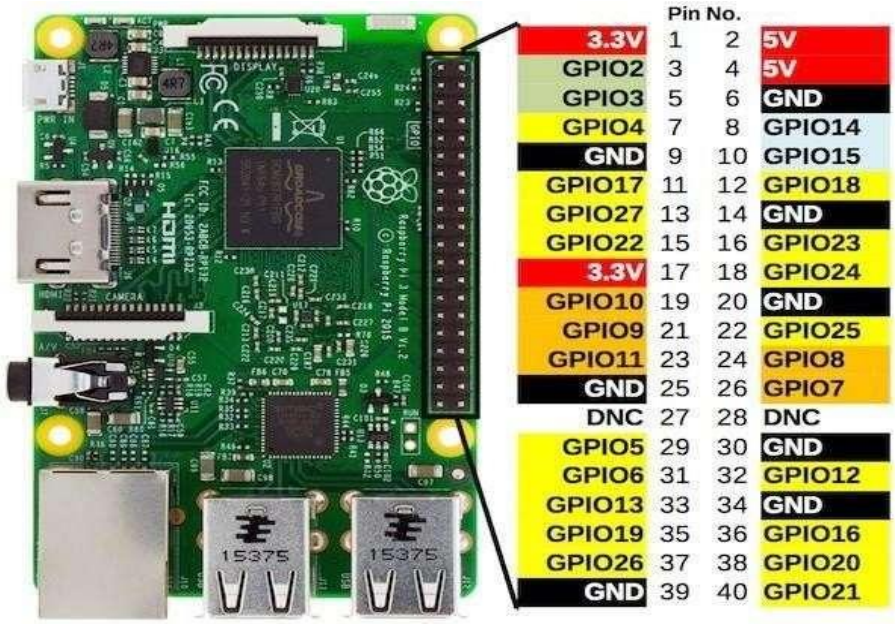

Fig 3.2: Raspberry Pi 3B

❖ **Specifications for Raspberry Pi 3 Model B:**

- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
- Dimensions 85.6mm x 56mm x 21mm
- 1GB RAM
- BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board
- 100 Base Ethernet
- 40-pin extended GPIO
- 4 USB 2 ports
- 4 Pole stereo output and composite video port
- Full size HDMI

- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- Micro SD port for loading your operating system and storing data
- Upgraded switched Micro USB power source up to 2.5A as shown in Fig 3.3 above

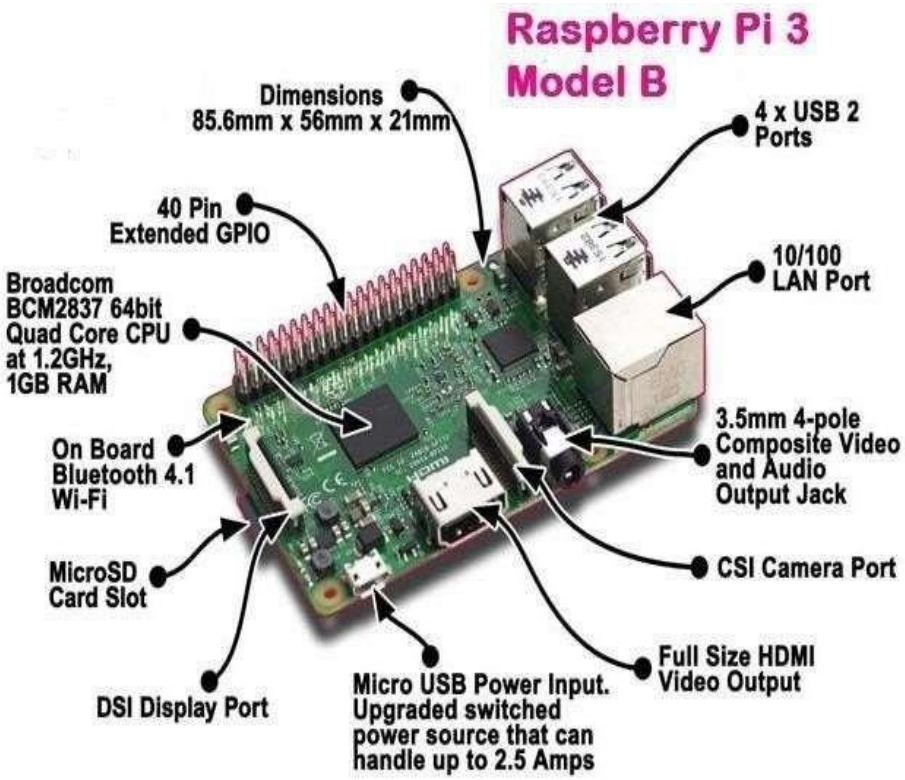

Fig 3.3: Specification of Raspberry Pi 3 Model B

3.3.2 MQ-3 Alcohol Sensor

The Grove - Gas Sensor (MQ3) module is useful for gas leakage detection (in home and industry). It is suitable for detecting Alcohol, Benzine, CH4, Hexane, LPG, CO. Due to its high sensitivity and fast response time, measurements can be taken as soon as possible. The sensitivity of the sensor can be adjusted by using the potentiometer as shown in the Fig 3.4.

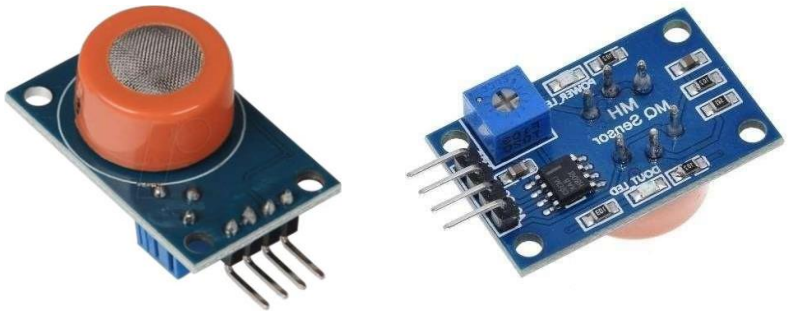

Fig 3.4: MQ-3 Alcohol Sensor

❖ Hardware Overview

This is an Analog output sensor. This needs to be connected to any one Analog socket in Grove Base Shield. The examples used in this tutorial makes uses of A0 analog pin. Connect this module to the A0 port of Base Shield.

It is possible to connect the Grove module to Arduino directly by using jumper wires by using the connection as shown below:

5V	VCC
GND	GND
NC	NC
Analog A0	SIG

The output voltage from the Gas sensor increases when the concentration of gas increases. Sensitivity can be adjusted by varying the potentiometer.

❖ Specifications of MQ-3 Alcohol Sensor

- Power requirements: 5 VDC @ ~165 mA (heater on) / ~60 mA (heater off)
- Current Consumption: 150mA
- DO output: TTL digital 0 and 1 (0.1 and 5V)
- AO output: 0.1- 0.3 V (relative to pollution), the maximum concentration of a voltage of about 4V
- Detecting Concentration: 0.05-10mg/L Alcohol
- Interface: 1 TTL compatible input (HSW), 1 TTL compatible output (ALR)
- Heater consumption: less than 750mW
- Operating temperature: 14 to 122 °F (-10 to 50°C)
- Load resistance: 200kΩ
- Sensitivity S: Rs(in air)/Rs(0.4mg/L Alcohol)≥5
- Sensing Resistance Rs: 2KΩ-20KΩ(in 0.4mg/l alcohol)
- Dimensions: 32 x 22 x 16 mm as shown in Fig 3.5 below,

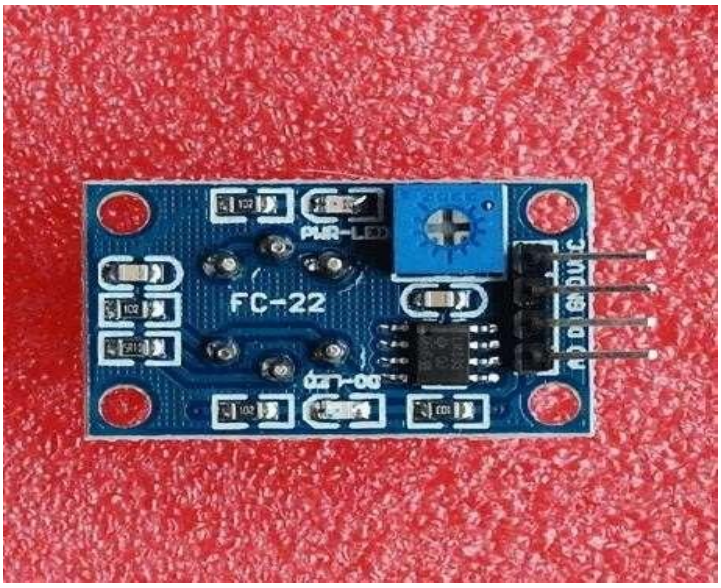

Fig 3.5: Specification of MQ-3 Alcohol Sensor

3.3.3 L298N Driver

The L298N is an integrated monolithic circuit in a 15- lead Multiwatt and PowerSO20 packages. It is a high voltage , high current dual full-bridge driver de-signed to accept standard TTL logic level sand drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the in-put signals .The emitters of the lower transistors of each bridge are connected together rand the corresponding external terminal can be used for the connection of an external sensing resistor as shown in Fig 3.6 below,

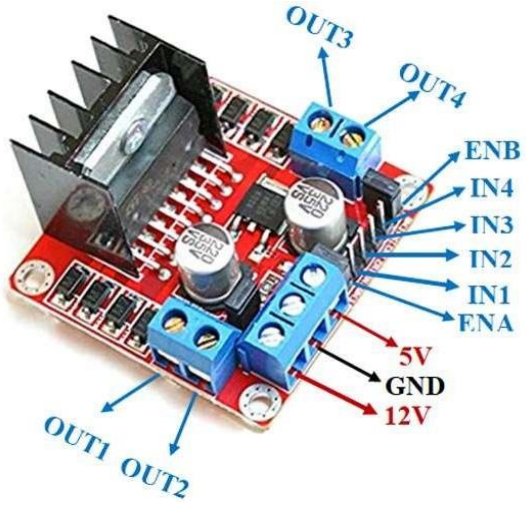

Fig 3.6: L298N Driver

❖ Functions of L298N Driver

- High operating voltage, which can be up to 40 volts.
- Large output current, the instantaneous peak current can be up to 3A.
- With 25W rated power.
- Two built in H-bridge, high voltage, large current, full bridge driver, which can be used to drive DC motors, stepper motors, relay coils and other inductive loads.
- Using standard logic level signal to control.
- Able to drive a two-phase stepper motor or four-phase stepper motor, and two-phase DC motors.
- Adopt a high-capacity filter capacitor and a freewheeling diode that protects devices in the circuit from being damaged by the reverse current of an inductive load, enhancing reliability.
- The module can utilize the built-in stability volt tube 78M05 to obtain 5v from the power supply. But to protect the chip of the 78M05 from damage, when the drive voltage is greater than 12v, an external 5v logic supply should be used.
- Drive voltage: 5-35V; logic voltage: 5V.
- PCB size: 4.2 x 4.2 cm ,as shown in Fig 3.7.

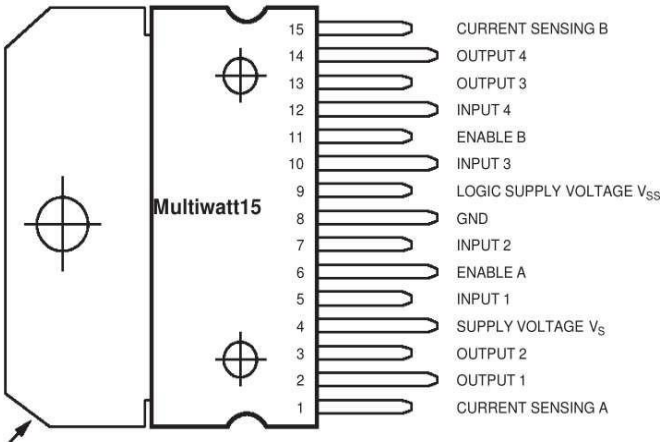

Fig 3.7: Functions of L298N Driver

3.3.4 BO Motor

Low density: lightweight, low inertia. Capability to absorb shock and vibration as a result of elastic compliance. Ability to operate with minimum or no lubrication, due to inherent lubricity. The relatively low coefficient of friction. Operating Voltage(VDC): 3~12 Shaft Length (mm): 8.5 Shaft Diameter (mm): 5.5 (Double D-type)No Load Current: 40-180mA. Rated Speed(After Reduction):100 RPM Rated Torque: 1 Kg cm as shown in Fig 3.8 below

Fig 3.8 : BO Motor

3.3.5 Web camera

A webcam is a video camera which is designed to record or stream to a computer or computer network. They are primarily used in video telephony, live streaming and social media, and security. Webcams can be built-in computer hardware or peripheral devices, and are commonly connected to a device using USB or wireless protocols. The webcam features are mainly dependent on the computer processor as well as an operating system of the computer. They can provide advanced features such as image archiving, motion sensing, custom coding, or even automation as shown in Fig 3.9 below,

Fig 3.9: Web Camera

3.4 SOFTWARE REQUIREMENTS

3.4.1 Geany

- Geany is an open-source, lightweight integrated development environment (IDE) that is designed for programmers. It provides a simple and intuitive interface with basic features for coding, making it a popular choice for beginners and those who prefer a less complex development environment.
- Geany is a free and open source editor/IDE as shown in the Fig 3.10. It is a great choice for both software development and text editing. It strikes a balance between the tech and non-tech worlds beautifully. It is highly customizable and available for Windows, Linux and Mac besides many other platforms.
- Geany is a great choice for a lightweight Integrated Development Environment for most programming languages. People who develop software using more than one programming language tend to like Geany for its programming language neutrality, customizability, light weight, multi-platform support, speed, stability and a very neat user interface.

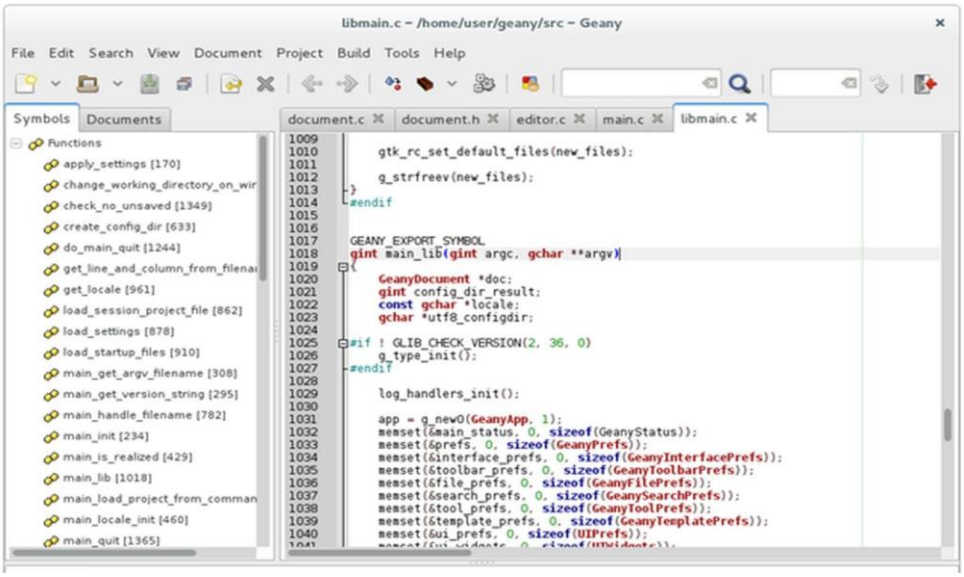

Fig : 3.10 Geany Software

❖ **Some of the features that make Geany an excellent IDE are:**

- Syntax highlighting
- Code navigation
- Code folding
- Auto-closing of XML and HTML tags
- Construct completion/snippets
- Symbol lists

3.4.2 Thonny

- Thonny is an integrated development environment for Python that is designed for beginners. It was created by Aivar Annamaa, an Estonian programmer as shown in the Fig 3.11. It supports different ways of stepping through code, step-by-step expression evaluation, detailed visualization of the call stack and a mode for explaining the concepts of references and heap.
- Thonny has received favourable reviews from Python and computer science education communities.
- It has been a recommended tool in several programming MOOCs. Since June 2017 it has been included by default in the Raspberry Pi's official operating system distribution Raspberry Pi OS.

❖ **Features of Thonny**

- Statement stepping without breakpoints
- Live variables during debugging
- Stepping through evaluation of the expressions (expressions get replaced by their values)
- Separate windows for executing function calls (for explaining local variables and call stack)
- Variables and memory can be explained either by using simplified model (name → value) or by using more realistic model (name → address/id → value)
- Simple pip GUI
- Support for CPython and MicroPython
- Support for running and managing files on a remote machine via SSH
- Possibility to log user actions for replaying or analyzing the programming process.

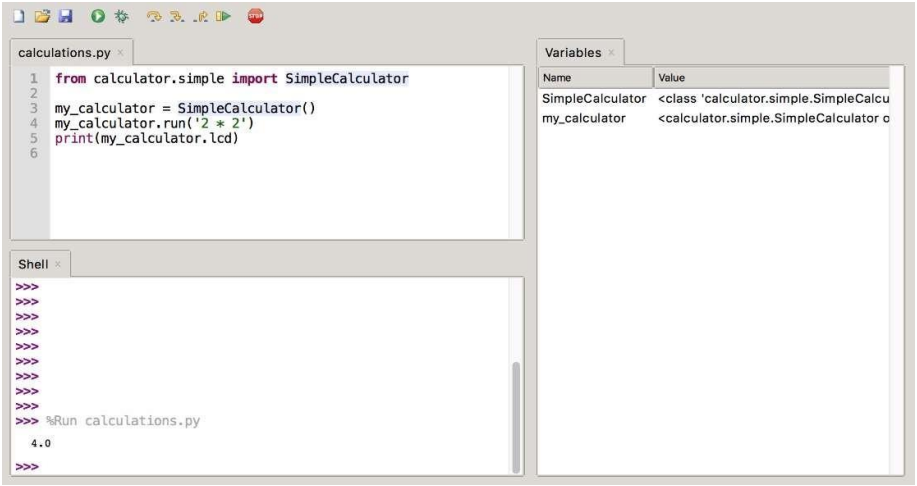

Fig : 3.11 Thonny Software

3.4.3 PyCharm

- Python is becoming popular among programmers who want to create software applications with short, tidy, and understandable code bases. By utilizing a variety of Python-integrated development environments (IDEs), they can even quicken the construction of unique software applications.
- PyCharm is a cross-platform IDE for Python created by JetBrains. PyCharm works with Windows, Linux, and macOS and supports Python versions 2. x and 3. x. At the same time, PyCharm's capabilities and features enable developers to create a variety of Python software applications rapidly and effectively.
- One of the most popular IDEs for the Python programming language is PyCharm. Large companies, including Twitter, Pinterest, HP, Symantec, and Groupon, already use Python IDE as shown in the Fig 3.12 below,

❖ PyCharm's Key Features

- Code Editor
- Code Navigation
- Refactoring
- Widely used web technologies
- Popular Python Web Frameworks are supported
- Scientific Python Libraries

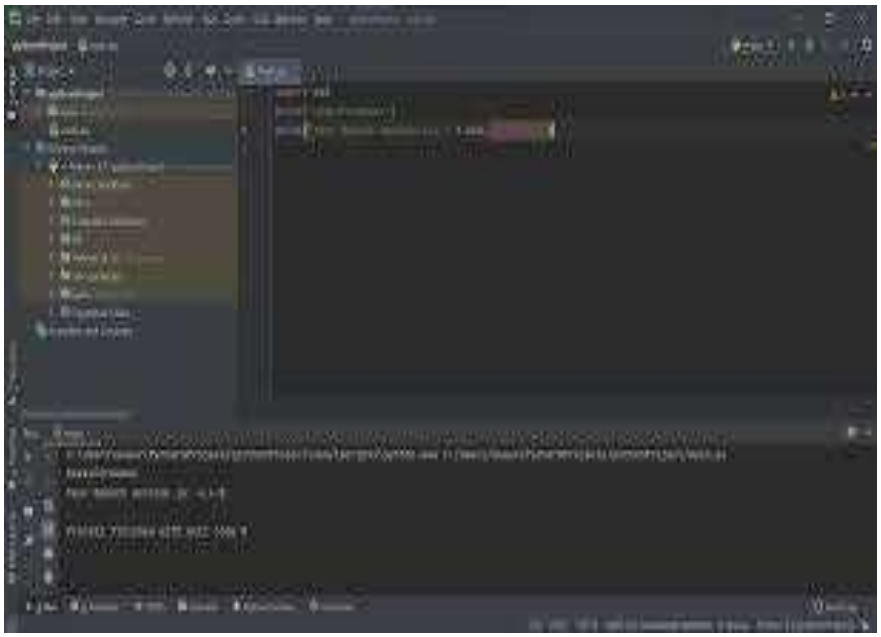

Fig : 3.12 PyCharm Software

3.4.4 Convolution Neural Network (CNN):

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning which takes an input image and assigns importance (weights and biases) to various features to help in distinguishing images.

A Neural Network is broadly classified into 3 layers:

- Input Layer
- Hidden Layer (can consist of one or more such layers)
- Output Layer

The Hidden layer can be further divided mainly into 2 layers:

- Convolution Layer: this extract features from the given input images.
- Full Connected Dense Layers: it assigns importance to the features from the convolution layer to generate output.

A convolutional neural network process generally involves two steps:

- Forward Propagation: the weights and biases are randomly initialized and this generates the output at the end
- Back Propagation: the weights and biases are randomly initialized in the beginning and depending on the error, the values are updated. Forward propagation happens with these updated values again and again for newer outputs to minimize the error. Activation functions are mathematical equations assigned to each neuron in the neural network and determine whether it should be activated or not depending on its importance (weight) in the image.

There are two types of activation functions:

- Linear Activation Functions: It takes the inputs, multiplied by the weights for each neuron, and creates an output signal proportional to the input. However, the problem with Linear Activation Functions is that it gives the output as a constant thus making it impossible for back propagation to be used as there is no relation to the input.
- Non-Linear Activation Functions: These create complex mappings between the different layers which help in better learning modelling of data. Few of the non-linear activation functions are - sigmoid, tanh, SoftMax, ReLU (Rectified Linear Units), Leaky ReLU etc.

❖ **Convolutional Neural Network (CNN) Model**

A proposed CNN architecture that includes convolutional layers, pooling layers and fully connected layers is illustrated in Fig 3.13. The total number of layers used to classify eye

state are ten layers. There are three convolutional and max pooling layers, one drop-out and fully connected layer. The convolutional layer will extract useful features from the input image. Each convolutional layer is connected to an RELU activation to increase the non-linear features in each image. Then, a max pooling layer is used, not only to preserve the main features, but also to reduce the size of the images. This helps CNN model reduce the amount of unimportant information while classifying the eye state image. Next, a dropout layer is deployed, this layer provides a way of combining many different neural network layers efficiently and prevents from overfitting. After That, a fully connected layer transforms the entire feature matrix received from the previous layers to a single column vector that represented the important features. After flattening, the single column will be processed by the RELU activation. This procedure increases the features non-linearity to enhance the softmax decision. Finally, a single neuron with sigmoid function is added for our binary classification (0,1 closed and open eye respectively).

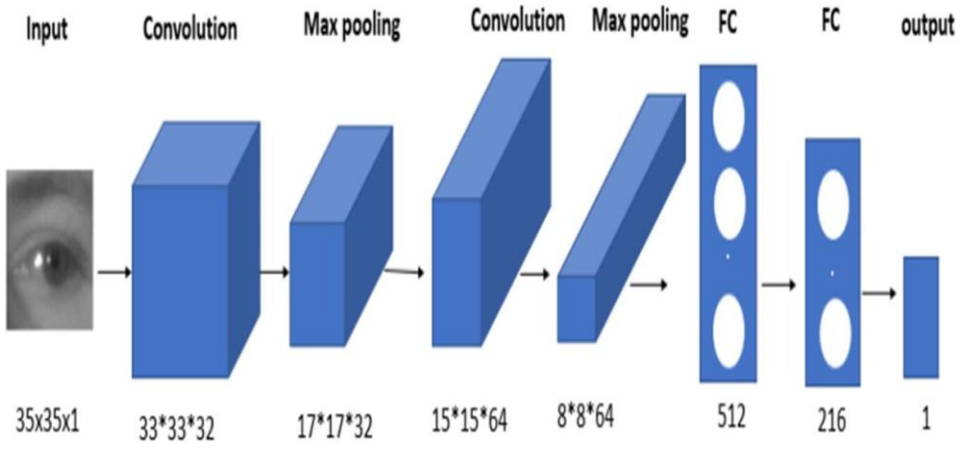

Fig : 3.13 Architecture Model Of Proposed CNN

❖ **Online Learning Process of CNN**

After the offline training process, we describe the online process of detecting the level of drowsiness as illustrated in Fig 3.14 First, we need to consider light condition adjustments. Then, features are extracted from the real time video input. These features are fed to the pre-train CNN model. In addition, Mouth Aspect Ratio (MAR) is used to evaluate symptoms of drowsiness from these features. As a result, a classification of the level of drowsiness is detected and an audio warning is announced to alert the drivers.

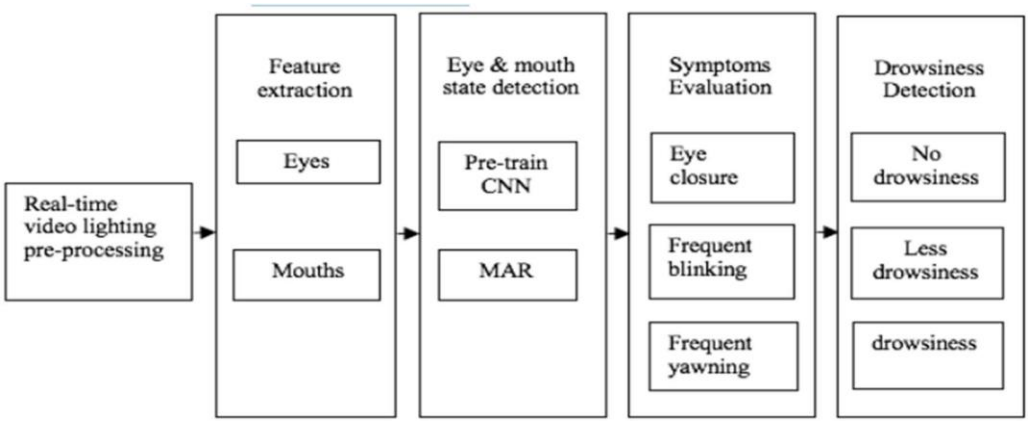

Fig : 3.14 Overview Of Online Learning Process

❖ **Feature Extraction**

Facial landmarks are used to localize and label salient regions of the face such as: eyes, eyebrows, nose, mouth, and jawline. The location of 68 coordinates (x, y) that maps to facial structure is shown in Fig 3.15. This paper considers only eye and mouth regions to detect driving drowsiness. To proceed, ensemble regression trees for facial landmark detection is used to detect the height and width eyes as reported in. To extract the area of the eye, 12 points are considered to find the eye bounding box. To do this, first, the face must be detected to extract the face bounding box. Face cascade is chosen because it is faster than other methods in terms of detection, tracking and efficiency. Figure 6b illustrates the approach to find the left eye bounding box ABCD. Here, p1, p2, p3, p4, p5 and p6 are 6 coordinates (x, y) eye landmark ratio. Following the same method as eye extraction, we use point 49 to point 62 as the mouth location.

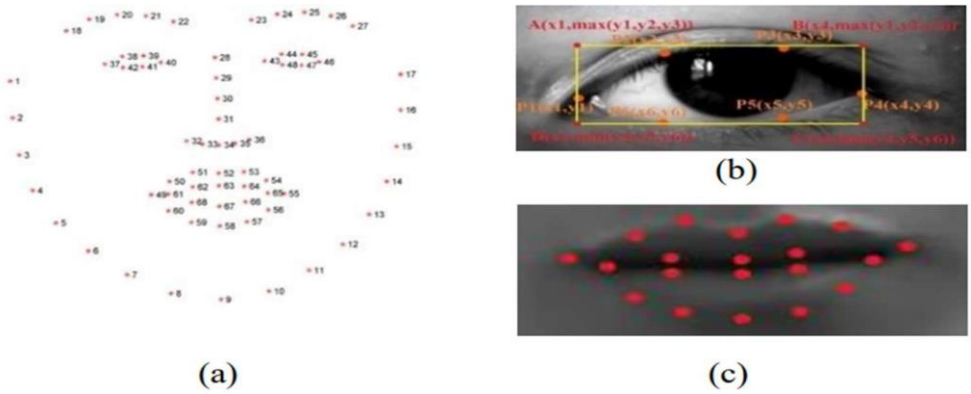

Fig : 3.15 (a) Visualizing 68 facial coordinate points, (b) Eye extraction, (c) Mouth extraction

❖ **Eye and Mouth State Detection**

After extracting both eye regions from real-time video, the left eye region (ABCD bounding box) is resized. This procedure is applied to fit the image input layer of the CNN model for training that is predefined to 35x35 pixels. These coordinators are calculated as follows:

Coordinate of Calibrated A = ($x1$+$x4$−35/2 , max ($y1$, $y2$, $y3$) - $x1$+$x4$−35/2)

Coordinate of Calibrated B = ($x1$+$x4$+35/2 , max ($y1$, $y2$, $y3$) - $x1$+$x4$+35/2)

Similarly, coordinate of calibrated D and C are calculated comparable to A and B respectively, with min (y4, y5, y6) and min (y1, y2, y3). As shown in Figure 3.19 points out of the 20 points are focused on selecting that were described in the previous paragraph. From these 6 points, the MAR (Mouth Aspect Ratio) equation is calculated as:

$MAR = |BF| + |CE| / 2 * |AD|$ (4)

Using the value of MAR, we can set the threshold to classify open/close mouth. According to our measurements of twelve volunteers, the average yawning threshold is set to 0.79. as illustrated in Fig 3.16 below,

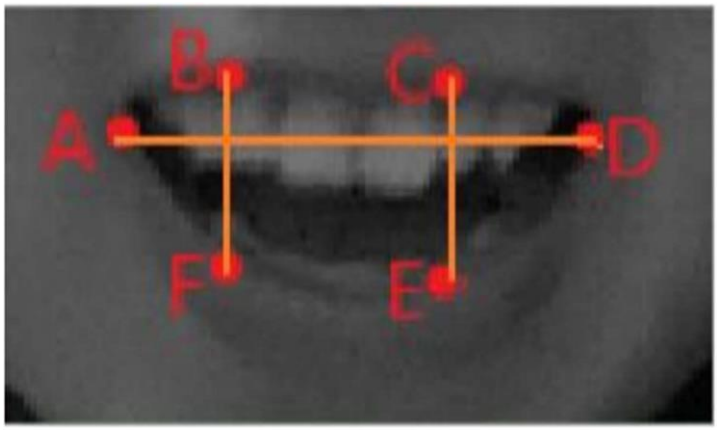

Fig : 3.16 Architecture Model Of Yawning

CHAPTER 4

DESIGN FLOW AND IMPLEMENTATION

4.1 Alcohol Detection with Ignition Control

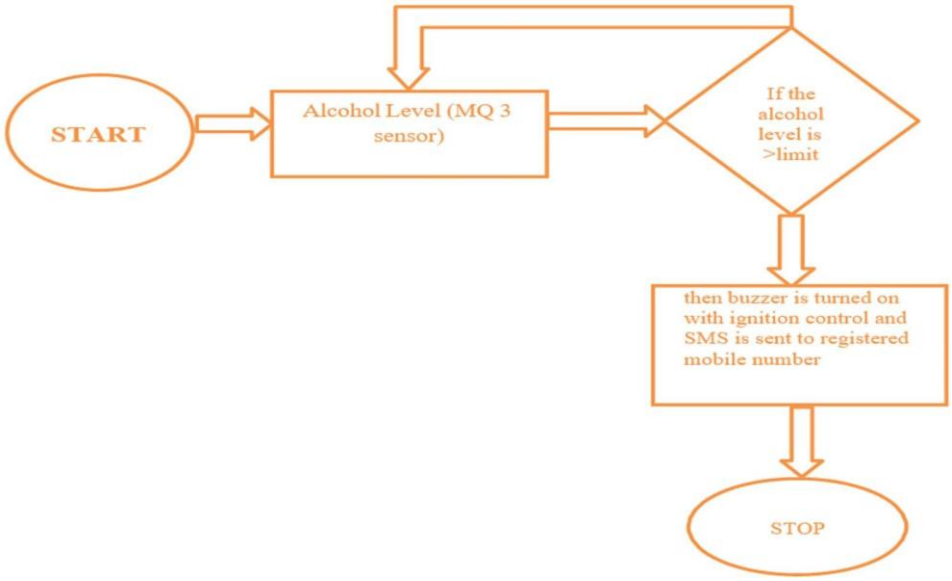

Fig : 4.1 Alcohol Detection with Ignition Control

The above Fig 4.1 represents the design flow of Alcohol Detection with Ignition Control purpose of this project is to detect drowsy, drunk and drive. We used CNN technique to detect drowsiness, MQ-3 sensor to monitor the alcohol level, if the alcohol level reaches beyond the permissible level, this proposed project will trigger, and the buzzer is turned on. If the alcohol does not exceedthe limit, then it is triggered back to the sensor with ignition control and sending a SMS to the respective registered mobile number.

4.2 Drowsiness Detection

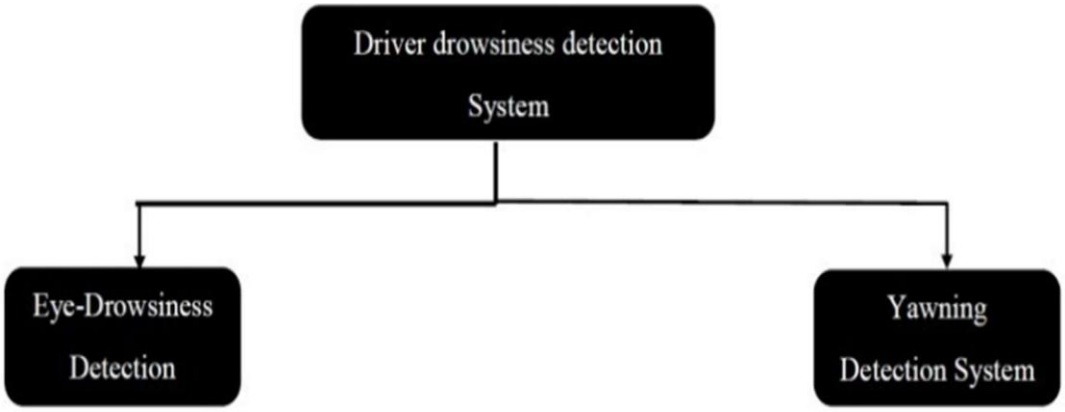

Fig : 4.2 Drowsiness Detection

The approaches we performed to detect the driver's level of drunkenness. trying to establish the eye-threshold value If these eye coordinates can be found, with P1, P4, and P2, P3, P5, and P6 being the intermediate locations in the eyelid and the extreme end points of the eye as shown in the above Fig 4.2 respectively, then the aspect ratio is determined as

EYE-ASPECT RATIO FORMULA= (|P2- P6|+|P3-P5|)2+|P1-P4|

The aspect ratio is continuously tracked, and when it falls below a specified point, the user is presumed to be drowsy

4.3 Eye-Drowsiness Detection System

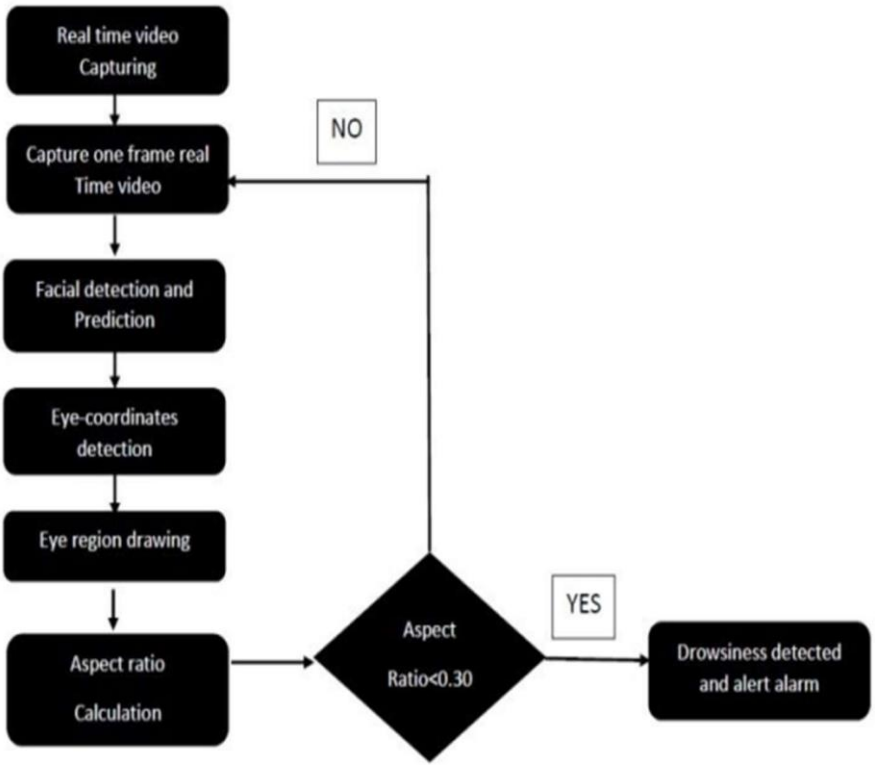

Fig 4.3 : Eye-Drowsiness Detection System

The above Fig 4.3 represents the design flow of Eye-Drowsiness Detection System, the driver is seated in front of the eye drowsiness detection system, which records real-time video of the driver. After that, it takes an image frame out of the video for subsequent editing. The driver's facial region is first assessed, then the driver's eye region. From the detected area, the eye surrounding coordinates are taken out and drawn for display in the output window. The aspect ratio of the eye is generated using these coordinates. The aspect ratio is then juxtaposed with a predetermined threshold value (0.30), below which the driver is identified as being sleepy and cautioned by an alarm, otherwise the procedure is repeated again.

4.4 Yawning Detection System

Fig 4.4 : Yawning Detection System

The above Fig 4.4 illustrates the Yawning Detection System, real-time collected video is also used by the yawning detection algorithm to extract image frames. Afterwards when, the face region is found and projected. The mouth and eye region, often known as facial landmarks, are marked and annotated. The midpoint annotation of the upper and lower lips is identified for the primary detection, and their Euclidean distance is calculated. The statement "Subject is Yawning" is printed and the yawn count is increased and shown if the Euclidean distance is above a specific threshold (25 mm); alternatively, the operation is repeated.

4.5 CODE IMPLEMENTATION

❖ **Code for Drowsiness Detection**

```
rom scipy.spatial import distance as dist
from imutils.video import VideoStream
from imutils import face_utils
from threading import Thread
import numpy as np
```

```
import argparse
import imutils
import time
import dlib
import cv2
import os

def  alarm(msg):
    global alarm_status
    global alarm_status2
    global saying

    while alarm_status:
        print('call')
        s = 'espeak "'+msg+'"'
        os.system(s)

    if alarm_status2:
        print('call')
        saying = True
        s = 'espeak "' + msg + '"'
        os.system(s)
        saying = False

def eye_aspect_ratio(eye):
    A = dist.euclidean(eye[1], eye[5])
    B = dist.euclidean(eye[2], eye[4])

    C = dist.euclidean(eye[0], eye[3])

    ear = (A + B) / (2.0 * C)

    return ear
```

```
def final_ear(shape):
    (lStart, lEnd) = face_utils.FACIAL_LANDMARKS_IDXS["left_eye"]
    (rStart, rEnd) = face_utils.FACIAL_LANDMARKS_IDXS["right_eye"]

    leftEye = shape[lStart:lEnd]
    rightEye = shape[rStart:rEnd]

    leftEAR = eye_aspect_ratio(leftEye)
    rightEAR = eye_aspect_ratio(rightEye)

    ear = (leftEAR + rightEAR) / 2.0
    return (ear, leftEye, rightEye)

def lip_distance(shape):
    top_lip = shape[50:53]
    top_lip = np.concatenate((top_lip, shape[61:64]))

    low_lip = shape[56:59]
    low_lip = np.concatenate((low_lip, shape[65:68]))

    top_mean = np.mean(top_lip, axis=0)
    low_mean = np.mean(low_lip, axis=0)

    distance = abs(top_mean[1] - low_mean[1])
    return distance
ap = argparse.ArgumentParser()
ap.add_argument("-w", "--webcam", type=int, default=0,
            help="index of webcam on system")
args = vars(ap.parse_args())

EYE_AR_THRESH = 0.3
EYE_AR_CONSEC_FRAMES = 30
YAWN_THRESH = 20
alarm_status = False
```

```
alarm_status2 = False
saying = False
COUNTER = 0

print("-> Loading the predictor and detector...")
#detector = dlib.get_frontal_face_detector()
detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")       #Faster but
less accurate
predictor = dlib.shape_predictor('shape_predictor_68_face_landmarks.dat')

print("-> Starting Video Stream")
vs = VideoStream(src=args["webcam"]).start()
#vs= VideoStream(usePiCamera=True).start()       //For Raspberry Pi
time.sleep(1.0)

while True:

   frame = vs.read()
   frame = imutils.resize(frame, width=450)
   gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

   #rects = detector(gray, 0)
   rects = detector.detectMultiScale(gray, scaleFactor=1.1,
                minNeighbors=5, minSize=(30, 30),
                flags=cv2.CASCADE_SCALE_IMAGE)

   #for rect in rects:
   for (x, y, w, h) in rects:
      rect = dlib.rectangle(int(x), int(y), int(x + w),int(y + h))

      shape = predictor(gray, rect)
      shape = face_utils.shape_to_np(shape)
      eye = final_ear(shape)
      ear = eye[0]
```

```
leftEye = eye [1]
rightEye = eye[2]

distance = lip_distance(shape)

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)

lip = shape[48:60]
cv2.drawContours(frame, [lip], -1, (0, 255, 0), 1)

if ear < EYE_AR_THRESH:
   COUNTER += 1

   if COUNTER >= EYE_AR_CONSEC_FRAMES:
      if alarm_status == False:
         alarm_status = True
         t = Thread(target=alarm, args=('wake up sir',))
         t.deamon = True
         t.start()

      cv2.putText(frame, "DROWSINESS ALERT!", (10, 30),
                  cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)

else:
   COUNTER = 0
   alarm_status = False

if (distance > YAWN_THRESH):
      cv2.putText(frame, "Yawn Alert", (10, 30),
                  cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
      if alarm_status2 == False and saying == False:
```

```
                alarm_status2 = True
                t = Thread(target=alarm, args=('take some fresh air sir',))
                t.deamon = True
                t.start()
            else:
                alarm_status2 = False

            cv2.putText(frame, "EAR: {:.2f}".format(ear), (300, 30),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
            cv2.putText(frame, "YAWN: {:.2f}".format(distance), (300, 60),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
        cv2.imshow("Frame", frame)
        key = cv2.waitKey(1) & 0xFF

        if key == ord("q"):
            break

    cv2.destroyAllWindows()
    vs.stop()
```

❖ **Code for Alcohol Detection With Ignition Control**

```
import RPi.GPIO as GPIO
import time
from twilio.rest import Client

 Set up the GPIO pins
account_sid = "AC9a9d3d3544431adf3d48f17b347c78aa"
auth_token  =  "9c4e449f3f15c91594f336ea93ed8b1b"
client = Client(account_sid, auth_token)
net = "your ward is drunk"
ALCOHOL_PIN = 18
ENA = 17
IN1 = 27
IN2 = 22
```

```
BUZZER = 21
LED = 26
ENB = 12
IN4 = 16
IN3 = 20
GPIO.setmode(GPIO.BCM)
GPIO.setup(ALCOHOL_PIN, GPIO.IN)
GPIO.setup(ENA,GPIO.OUT)
GPIO.setup(IN1,GPIO.OUT)
GPIO.setup(IN2,GPIO.OUT)
GPIO.setup(ENB,GPIO.OUT)
GPIO.setup(IN3,GPIO.OUT)
GPIO.setup(IN4,GPIO.OUT)
GPIO.setup(BUZZER,GPIO.OUT)
GPIO.setup(LED,GPIO.OUT)
def forward():
   GPIO.output(ENA,GPIO.HIGH)
   GPIO.output(IN1,GPIO.HIGH)
   GPIO.output(IN2,GPIO.LOW)
   GPIO.output(ENB,GPIO.HIGH)
   GPIO.output(IN3,GPIO.LOW)
   GPIO.output(IN4,GPIO.HIGH)
def stop():
   GPIO.output(ENA,GPIO.LOW)
   GPIO.output(IN1,GPIO.LOW)
   GPIO.output(IN2,GPIO.LOW)
   GPIO.output(ENB,GPIO.LOW)
   GPIO.output(IN3,GPIO.LOW)
   GPIO.output(IN4,GPIO.LOW)
# Main loop
while True:
   # Read the digital signal from the alcohol sensor
   alcohol_value = GPIO.input(ALCOHOL_PIN)
   # Check the alcohol concentration level and take appropriate action
```

```
  if alcohol_value != 0:
      forward()
      GPIO.output(BUZZER,GPIO.LOW)
      GPIO.output(LED,GPIO.LOW)
      print("No alcohol detected")
  else:
      stop()
     message                                                                    =
    client.api.account.messages.create(to="+917760355680",from_="+14407501797",
    body=net)
      GPIO.output(BUZZER,GPIO.HIGH)
      GPIO.output(LED,GPIO.HIGH)
      print("Alcohol detected!")

  # Wait for a short time before taking the next measurement
  time.sleep(0.1)
# for buzzer
    # Check the alcohol concentration level and take appropriate action
```

CHAPTER 5

RESULTS AND CONCLUSION

5.1 Drowsiness and Yawning Detection

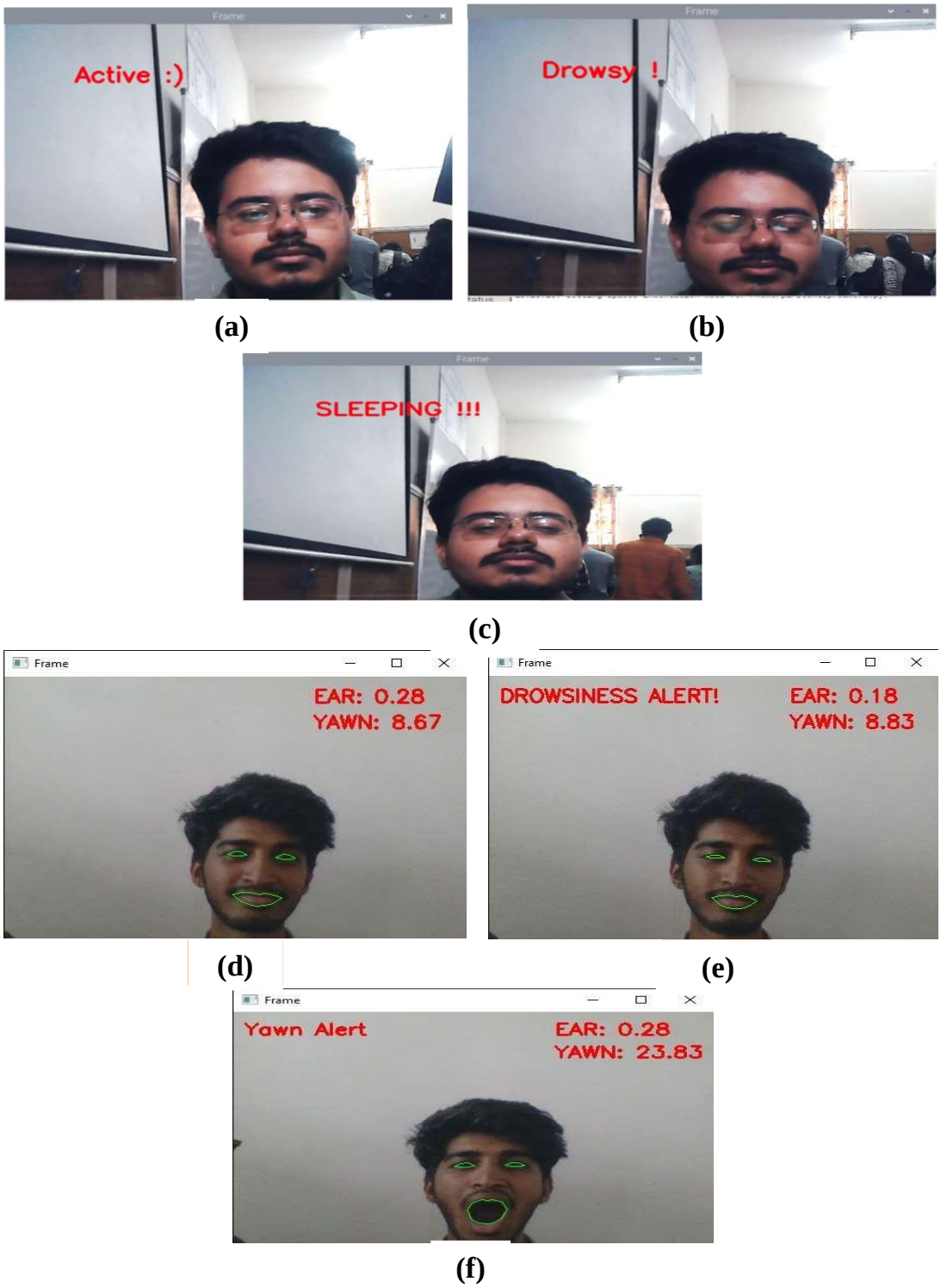

Fig 5.1 (a), (b), (c), (d), (e) & (f) : Drowsiness And Yawning Detection

5.2 Alcohol Detection with Ignition Control

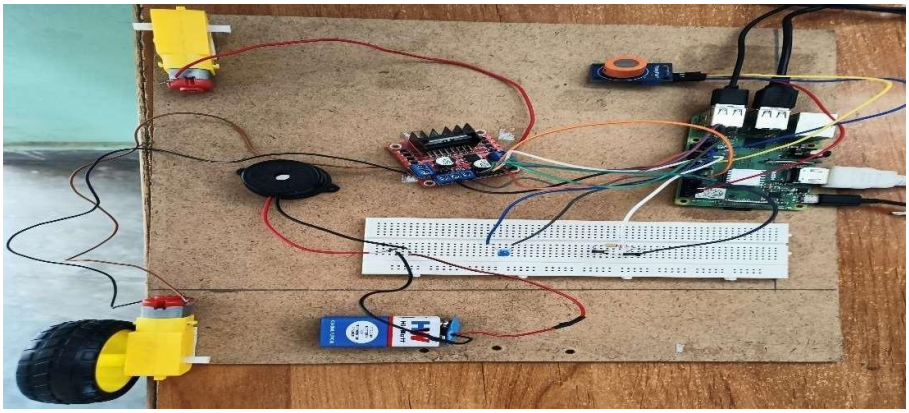

Fig : 5.2 Output of Alcohol Detection with Ignition Control

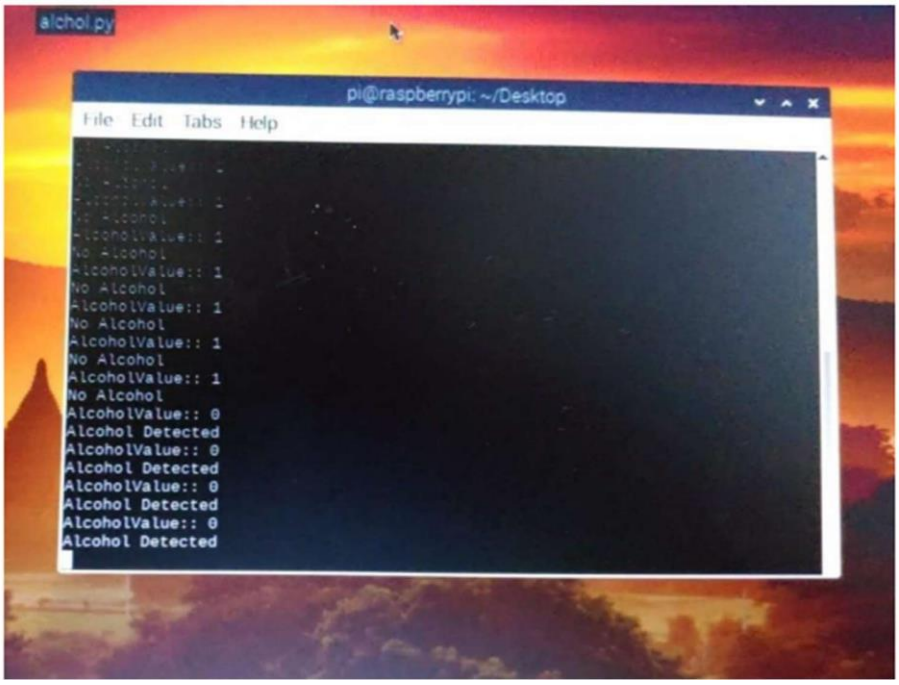

Fig : 5.3 Output Window of Alcohol Detection

From the above Figures 5.1, 5.2 & 5.3 represents the design flow of Drowsiness and Alcohol Detection with Ignition Control, the purpose of this project is to detect drowsy, drunk and drive. We used CNN (Convolutional Neural Network) technique to detect drowsiness (eye and yawning detection), MQ-3 sensor to monitor the alcohol level, if the alcohol level reaches beyond the permissible level, this proposed project will trigger, and the buzzer is turned on. If the alcohol does not exceed the limit, then it is triggered back to the sensor with ignition control and sending a SMS to the respective registered mobile number.

CHAPTER 6

APPLICATIONS

❖ Mining industries

A mouthpiece is attached to the breathalyzer, that allows the workers to blow into the device. During a routine breath test, the breath passes through the vessel to the measurement component of the detector which detects the level of alcohol content.

❖ Aviation

In airports a mouthpiece is attached to the breathalyzer, that allows the pilots to blow into the device. During a routine breath test, the breath passes through the vessel to the measurement component of the detector which detects the level of alcohol content of the pilots.

❖ Railways

In the railways the loco pilots entrusted with the job of operating passenger and goods trains have to take breath analyzing test before and after the train journey. The existing breath analyzers are either over-aged or faulty and should be upgraded immediately.

❖ Army Vehicles

In the army vehicles the driver at the driving seat, has to blow towards the alcohol sensor on the device to start the vehicle. This will immediately activate the sensor that will analyze and estimate the alcohol content present in the blood of the driver.

REFERENCES

[1] Supraja , Bhanu Sri, Mary Posonia "Drunk and Drive Detection with Ignition Control using Iot" International Journal of Pure and Applied Mathematics, Volume 118 No. 20 2022.

[2] T. Venkat Narayana Rao, Kartik Reddy Yellu. "Preventing Drunken Driving Accidents using IOT" International Journal of Advanced Research in Computer Science, Volume 8, No. 3, March – April 2022.

[3] Vaishnavi. M , Umadevi.V , Vinothini. M , Bhaskar Rao . Y , Pavithra. S "Intelligent Drowsiness and Alcohol Detection System For Car" International Journal of Scientific & Engineering Research, Volume 5, Issue 11, November-2021.

[4] Ms.Shruthi S.Gujarathi, Mr.Gaurav Ahirrao "Drowsiness Detection and Automatic Drunken Drive Avoiding System" in IJERA on April 2021.

[5] Rajesh Kumar Jakkar, Roop Pahuja, Raj Kumar Saini, Bhagirath Sahu, Natwar "Drunk, Drive and Drowsy Detection and Alert system for vehicles" in American journal of Traffic and Transportation Engineering on August 10, 2021.

[6] A.A. Daundkar, S.A. DhaigudePatil, Shweta P.Bidgar "Mobile Phone Based Drowsiness and Drunk Drive Detection" in IJRCS on March 2020.

[7] Aditya Ranjan, Karan Vyas, Sujay Ghadge, Siddharth Patel, Suvarna Sanjay Pawar, "Driver Drowsiness Detection System Using Computer Vision.", in International Research Journal of Engineering and Technology(IRJET), 2020.