

**PROJECT BASED REPORT**  
**on**  
**DROWSINESS DETECTION AND ALERTING SYSTEM**

**Submitted in partial fulfilment of the**  
**Requirements for the award of the Degree of**  
**Bachelor of Technology**

**in**  
**Computer Science and Engineering**

**Under the esteemed guidance of**

**Mr. D. Sandeep**  
**Project Coordinator**

**by**

**G. S. S. HARSHA**  
**B. SWARNA KUMAR**  
**K. HASWANTH**



## **ACKNOWLEDGEMENT**

Our sincere thanks to **D. Sandeep** in the Lab for his outstanding support throughout the project for the successful completion of the work.

We express our gratitude to **D. Sandeep**, Project coordinator for providing us with adequate facilities, ways and means by which we can complete this project-based Lab.

Last but not the least, we thank all Teaching and Non-Teaching Staff of our department and especially our classmates and our friends for their support in the completion of our project-based Lab.

**G.S.S. HARSHA**

**B. SWARNA KUMAR**

**K. HASWANTH**

## TABLE OF CONTENTS

CHAPTERS	PAGE NO
CHAPTER 1: INTRODUCTION	5
1.1 INTRODUCTION	5
1.2 PROBLEM DEFINITION	5
1.3 SCOPE	5
1.4 PURPOSE	6
1.5PROBLEM AND EXISTING TECHNOLOGY	6
1.6 PROPOSED SYSTEM	6
CHAPTER 2: REQUIREMENTS & ANALYSIS	7
2.1 PLATFORM REQUIREMENTS	7
2.2 PROJECT ANALYSIS	7
CHAPTER 3: DESIGN & IMPLEMENTATION	9
3.1FLOWCHART	9
3.2 PSEUDO CODE	10
CHAPTER 4: SCREENSHOTS	16
CHAPTER 5: CONCLUSION	20
CHAPTER 6: FUTURE SCOPE	21
CHAPTER 7: REFERENCES	22

## **ABSTRACT**

A majority of road related accidents are due to driver's fatigue, distraction and drowsiness. Recently accidents are increased at large amount. Various new creative technologies are introduced to avoid and reduce these accidents. In these accidents, driver can meet severe injuries and even death. These may lead to significant economic loss. This system can reduce the road related accidents due to driver's fatigue. The most widely recognized strategy to distinguish the driver's state is of utilizing face detection. primarily focuses on the edges of the face through which the weakness condition is obtained. tiredness, stress, consumption of alcohol, mobile utilization etc. Our main objective is to develop a model which can detect the driver's drowsiness and indicate the driver's condition by alarm. Here we provide a prevention technique using eye blink and where we can stop the vehicle to avoid collision of vehicle. In the existing system, the drowsiness is detected by the SVM (Support Vector Machine). This SVM classifies a sequence of video segments into alert or non-alert driving event. To overcome the drawbacks of the existing system, the proposed system implements, Eye aspect ratio (EAR) algorithm for drowsiness detection is used.

**KEYWORDS:** Drowsiness Detection, Eye blink Detection, Driver's Fatigue Condition.

## **1.INTRODUCTION**

### **1.1. Introduction**

Drowsiness is a condition which diminishes the awareness caused by the absence of rest or weariness. Because of weariness, driver lose their control which may divert them from the street and prompts serious mishaps. Basically, these mishaps may because of the driver tired condition. Driving continuously for quite a while prompts tiredness and make them to lost their awareness. Due to substantial increment in the number of mischances day by day which makes a major issues. Therefore, a model is required to keep the driver in the engaged state. The goal of our model is to make a driver weariness location which demonstrates driver's lethargic condition through their face. The thought is to build up the model that will persistently recognize the driver's exhaustion progressively. One of the identification procedure is Percentage of squinting of eyes. Here we utilize a system totally in view of the development of eyes. The perception of eye development is essentially in view of the driver's condition. On the off chance that the individual is expending liquor it can be effortlessly identified by alcoholic sensors and if any temperature misused it can be recognized by temperature sensors. The spillage of gas can likewise be distinguished by the gas sensors. In request to identify the driver's laziness different methods has been implemented. But no strategy has been actualized to stop the auto naturally if the weakness or sluggish mode is identified. The most widely recognized strategy to distinguish the driver's state is of utilizing face detection. primarily focuses on the edges of the face through which the weakness condition is obtained. tiredness, stress, consumption of alcohol, mobile utilization etc.

### **1.2 Problem Definition**

Drowsiness and Fatigue of drivers are amongst the significant causes of road accidents. Every year, they increase the amounts of deaths and fatalities injuries globally. This is a computer vision system that can automatically detect person drowsiness in a real-time video stream and then play an alarm if the driver appears to be drowsy.

### **1.3 Scope**

This application is basically written as a solution to the drawbacks of existing system. This application can be used as a real-world application and by any organization. It's could be used as a general application with few minor modifications.

## **1.4 Purpose**

The purpose of this application is to develop a device which is a web application which provides a service to all the employees. The problem with the existing system is that the drowsiness is detected by the SVM (Support Vector Machine). This SVM classifies a sequence of video segments into alert or non-alert driving event. Visual intelligence and Artificial Intelligence are being used in various systems. This VI and AI detects the facial recognition and eye-blinking using image processing. Here the cost to develop the project will be less but the reliability will be degraded. Thus, this application provides a solution to problems of drowsiness of drivers by alerting them with a sound.

## **1.5 Problem and Existing Technology**

- ❖ In the existing system, the drowsiness is detected by the SVM (Support Vector Machine).
- ❖ This SVM classifies a sequence of video segments into alert or non-alert driving event.
- ❖ Visual intelligence and Artificial Intelligence are being used in various systems.
- ❖ This VI and AI detects the facial recognition and eye-blinking using image processing.

## **1.6 Proposed system**

- ❖ In the proposed system, Eye aspect ratio (EAR) algorithm for drowsiness detection is used.
- ❖ Sending message to the respective authorities to inform his/her drowsiness condition in the form of speech.
- ❖ Continuous live video monitoring of driver to prevent accidents by alerting the driver using the audio speech.
- ❖ Making use of mobile app to detect the driver's alertness condition.

## 2.REQUIREMENTS & ANALYSIS

### 2.1. Platform Requirements

Hardware/ Software	Hardware / Software element	Specification /version
Hardware	Processor	Intel core to duo
	RAM	1 GB
	Hard Disk	100 GB
	NODEMCU, BASIC SHEILD, CAMERA LOGITECH	
Software	OS	Windows 10
	Python IDLE	
	ANACONDA	
	IBM CLOUD	
	ARDUINO	

**Tab 2.1. Platform Requirements**

### 2.2 Project Analysis

- ❖ Drowsiness and Fatigue of drivers are amongst the significant causes of road accidents.
- ❖ Every year, they increase the amounts of deaths and fatalities injuries globally.
- ❖ This is a computer vision system that can automatically detect person drowsiness in a real-time video stream and then play an alarm if the driver appears to be drowsy.
- ❖ This system can be implemented in the field where we need to test the alertness of the person.

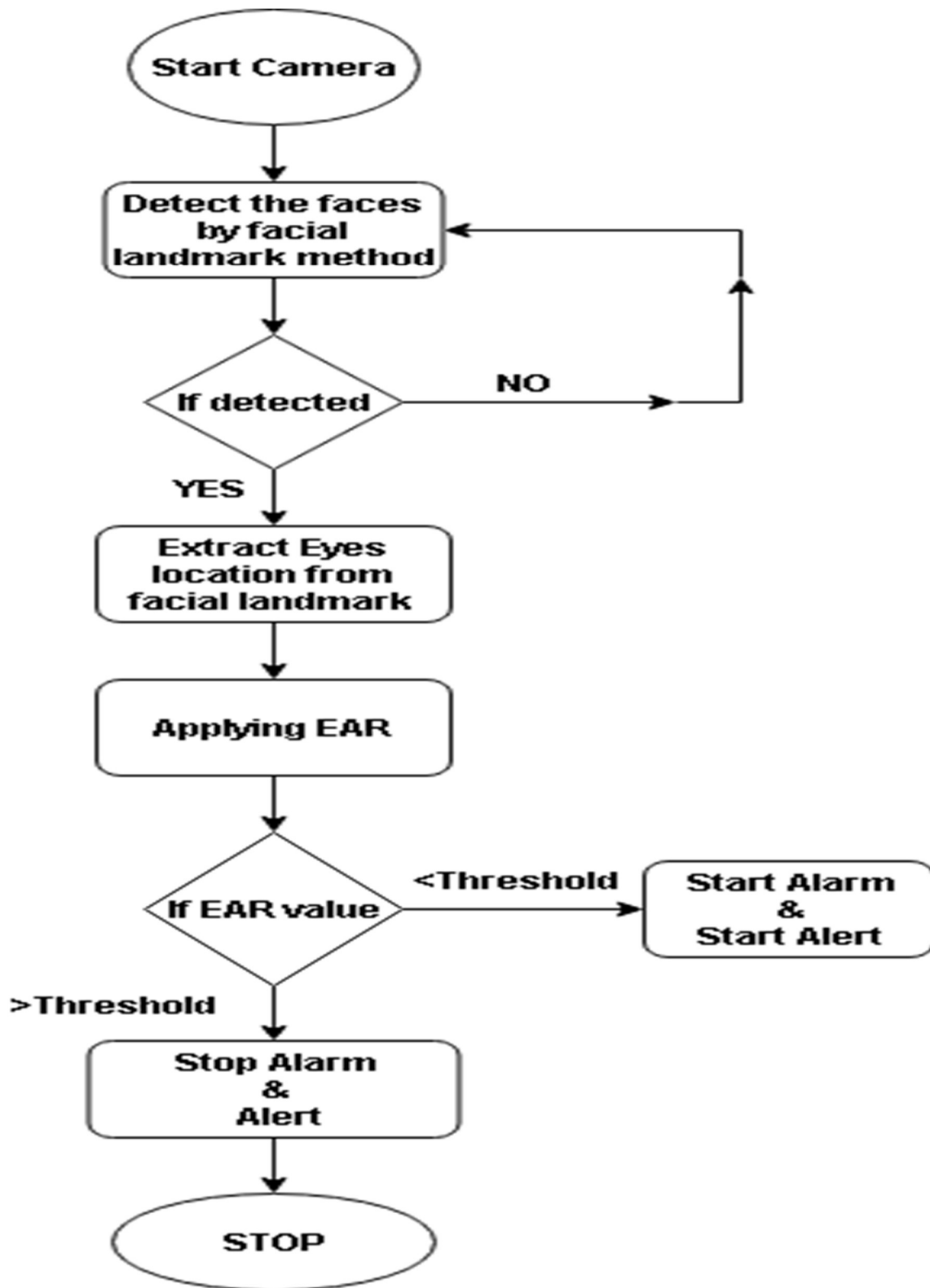
- ❖ Once we have our eye regions, we can apply the eye aspect ratio to determine if the eyes are closed. If the eyes have been closed for a sufficiently long enough period, we can assume the user is at risk of falling asleep and sound an alarm to grab their attention.
- ❖ Eye aspect ratio algorithm is used to detect whether the person is drowsy or not.
- ❖ There will be the video streaming and from that When we detect that the person is drowsy or sleeping, we can alert him through some sound alerts.
- ❖ We can keep the desired text in the form of message and convert that message to speech.
- ❖ We can inform the authorities when the worker is drowsy or sleeping by sending them messages, so that he can take necessary actions.
- ❖ Using Eye- aspect ratio algorithm for drowsiness detection
- ❖ Sending Message to the respective authorities to inform his drowsiness condition in the form of speech
- ❖ Continuous live video monitoring of driver to prevent accidents.
- ❖ Making use of Mobile App to detect Driver's alertness condition

### **Drowsiness Detection Algorithm:**

- ❖ Firstly, setting up the camera that monitors a stream of faces.
- ❖ If a face is found, we apply facial landmark detection and extract the eye regions. Now that we have the eye regions, we can compute the eye aspect ratio to determine if the eyes are closed.
- ❖ If the eye aspect ratio indicates that the eyes have been closed for a sufficiently long enough amount of time, we'll sound an alarm to wake up the driver.



### 3.DESIGN AND IMPLEMENTATION



## 3.2 Source Code

```
# USAGE

# python detect_drowsiness.py --shape-predictor shape_predictor_68_face_landmarks.dat
# python detect_drowsiness.py --shape-predictor shape_predictor_68_face_landmarks.dat --
alarm alarm.wav


# import the necessary packages
from 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 playsound
import argparse
import imutils
import time
import dlib
import requests
import cv2

def sound_alarm(path):
    # play an alarm sound
    playsound.playsound(path)


def eye_aspect_ratio(eye):
    # compute the euclidean distances between the two sets of
    # vertical eye landmarks (x, y)-coordinates
    A = dist.euclidean(eye[1], eye[5])
    B = dist.euclidean(eye[2], eye[4])

    # compute the euclidean distance between the horizontal
    # eye landmark (x, y)-coordinates
```

```

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

# compute the eye aspect ratio
ear = (A + B) / (2.0 * C)

# return the eye aspect ratio
return ear

# construct the argument parse and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-p", "--shape-predictor", required=True,
help="path to facial landmark predictor")
ap.add_argument("-a", "--alarm", type=str, default="",
help="path alarm .WAV file")
ap.add_argument("-w", "--webcam", type=int, default=0,
help="index of webcam on system")
args = vars(ap.parse_args())

# define two constants, one for the eye aspect ratio to indicate
# blink and then a second constant for the number of consecutive
# frames the eye must be below the threshold for to set off the
# alarm
EYE_AR_THRESH = 0.3
EYE_AR_CONSEC_FRAMES = 48

# initialize the frame counter as well as a boolean used to
# indicate if the alarm is going off
COUNTER = 0
ALARM_ON = False

# initialize dlib's face detector (HOG-based) and then create

```

```

# the facial landmark predictor
print("[INFO] loading facial landmark predictor...")
detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor(args["shape_predictor"])

# grab the indexes of the facial landmarks for the left and
# right eye, respectively
(lStart, lEnd) = face_utils.FACIAL_LANDMARKS_IDXS["left_eye"]
(rStart, rEnd) = face_utils.FACIAL_LANDMARKS_IDXS["right_eye"]

# start the video stream thread
print("[INFO] starting video stream thread...")
vs = VideoStream(src=args["webcam"]).start()
time.sleep(1.0)

# loop over frames from the video stream
while True:
    # grab the frame from the threaded video file stream, resize
    # it, and convert it to grayscale
    # channels)
    frame = vs.read()
    frame = imutils.resize(frame, width=450)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # detect faces in the grayscale frame
    rects = detector(gray, 0)

    # loop over the face detections
    for rect in rects:
        # determine the facial landmarks for the face region, then
        # convert the facial landmark (x, y)-coordinates to a NumPy

```

```

# array
shape = predictor(gray, rect)
shape = face_utils.shape_to_np(shape)

# extract the left and right eye coordinates, then use the
# coordinates to compute the eye aspect ratio for both eyes
leftEye = shape[lStart:lEnd]
rightEye = shape[rStart:rEnd]
leftEAR = eye_aspect_ratio(leftEye)
rightEAR = eye_aspect_ratio(rightEye)

# average the eye aspect ratio together for both eyes
ear = (leftEAR + rightEAR) / 2.0

# compute the convex hull for the left and right eye, then
# visualize each of the eyes
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)

# check to see if the eye aspect ratio is below the blink
# threshold, and if so, increment the blink frame counter
if ear < EYE_AR_THRESH:
    COUNTER += 1

# if the eyes were closed for a sufficient number of
# then sound the alarm
if COUNTER >= EYE_AR_CONSEC_FRAMES:
    requests.get('https://con-node.eu-gb.mybluemix.net/data?command=Alert')
# if the alarm is not on, turn it on

```

```

if not ALARM_ON:
    ALARM_ON = True

# check to see if an alarm file was supplied,
# and if so, start a thread to have the alarm
# sound played in the background
if args["alarm"] != "":
    t = Thread(target=sound_alarm,
    args=(args["alarm"],))
    t.daemon = True
    t.start()

# draw an alarm on the frame
cv2.putText(frame, "DROWSINESS ALERT!", (10, 30),
cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)

# otherwise, the eye aspect ratio is not below the blink
# threshold, so reset the counter and alarm
else:
    requests.get('https://con-node.eu-gb.mybluemix.net/data?command=LIGHTOFF')
    COUNTER = 0
    ALARM_ON = False

# draw the computed eye aspect ratio on the frame to help
# with debugging and setting the correct eye aspect ratio
# thresholds and frame counters
cv2.putText(frame, "EAR: {:.2f}".format(ear), (300, 30),
cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)

# show the frame
cv2.imshow("Frame", frame)

```

```
key = cv2.waitKey(1) & 0xFF
```

```
# if the `q` key was pressed, break from the loop
```

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

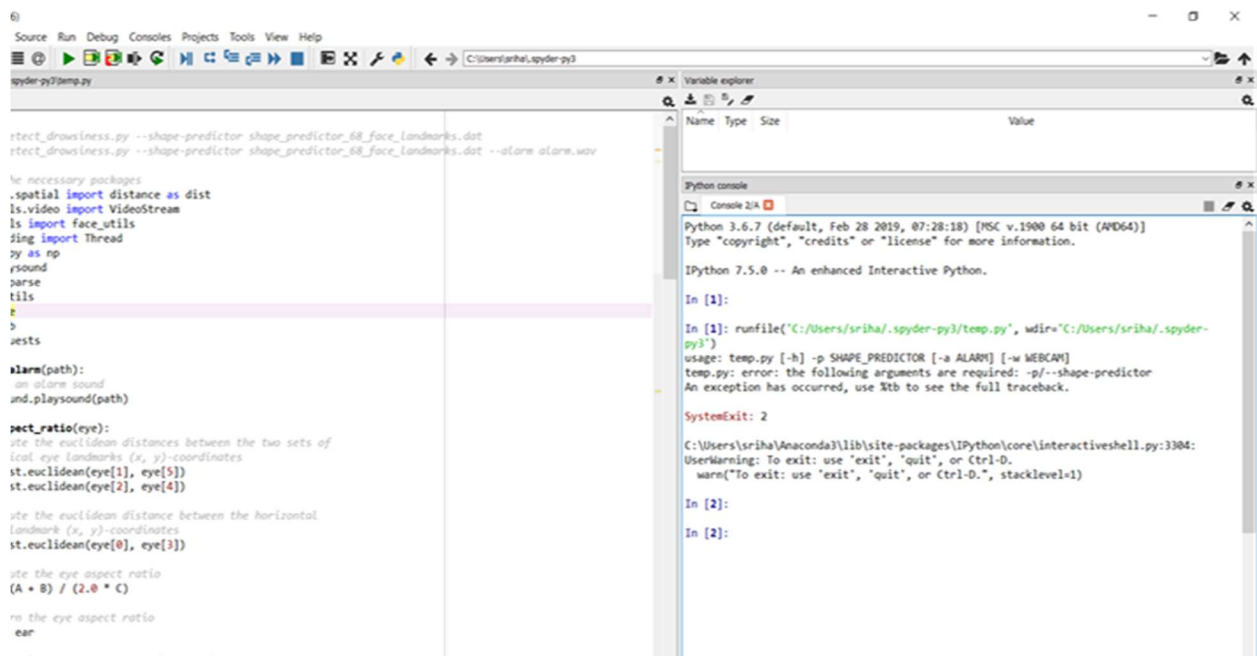
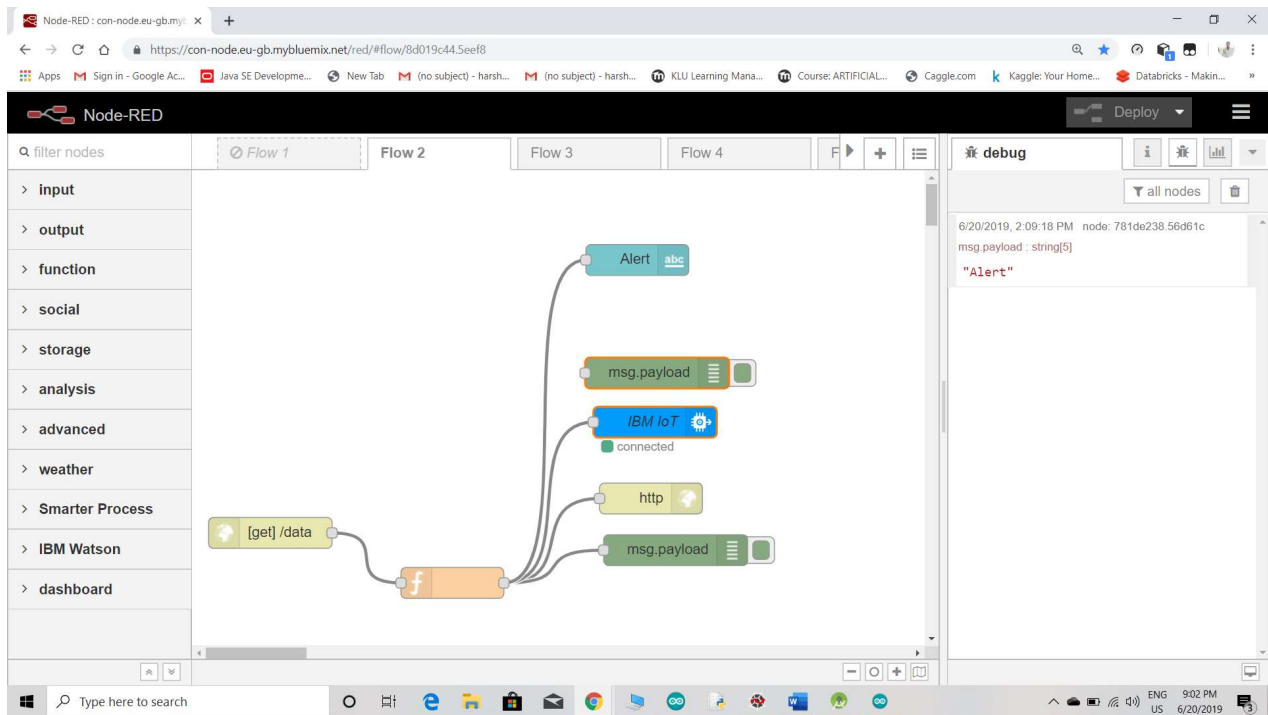
```
    break
```

```
# do a bit of cleanup
```

```
cv2.destroyAllWindows()
```

```
vs.stop()
```

## 4.SCREENSHOTS





```
SmartHomeControl | Arduino 1.8.9
File Edit Sketch Tools Help

SmartHomeControl

#include <ESP8266WiFi.h>
#include <PubSubClient.h>

const char* ssid = "haswanth";
const char* password = "iwontsay";

#define ORG "j533y3"
#define DEVICE_TYPE "Drowsiness"
#define DEVICE_ID "123456789"
#define TOKEN "123456789"
String command;
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char topic[] = "iot-2/cmd/home/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
//Serial.println(clientId);
void callback(char* topic, byte* payload, unsigned int payloadLength);
WiFiClient wifiClient;
PubSubClient client(server, 1883, callback, wifiClient);

void setup() {
  Serial.begin(115200);
  Serial.println();
  pinMode(D1, OUTPUT);
  wifiConnect();
  mqttConnect();
}

void loop() {

Done uploading.
Invalid library found in C:\Users\swarna\Documents\Arduino\libraries\libraries: no headers files (.h) found in C:\Users\swarna\Documents\Arduino\libraries\libraries
Invalid library found in C:\Users\swarna\Documents\Arduino\libraries\libraries: no headers files (.h) found in C:\Users\swarna\Documents\Arduino\libraries\libraries

79 NodeMCU 1.0 (ESP-12E Module) on COM18
15:07
20-06-2019
```

```
SmartHomeControl | Arduino 1.8.9
File Edit Sketch Tools Help

SmartHomeControl

#include <ESP8266WiFi.h>
#include <PubSubClient.h>

const char* ssid = "haswanth";
const char* password = "iwontsay";

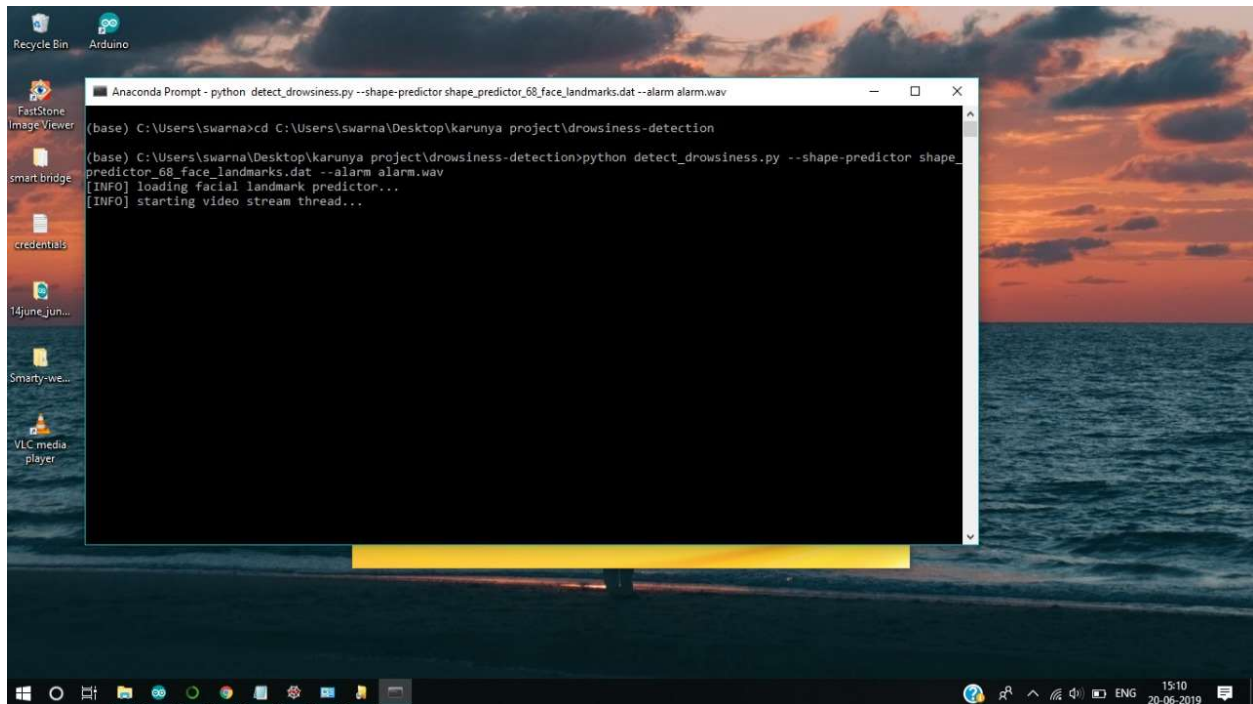
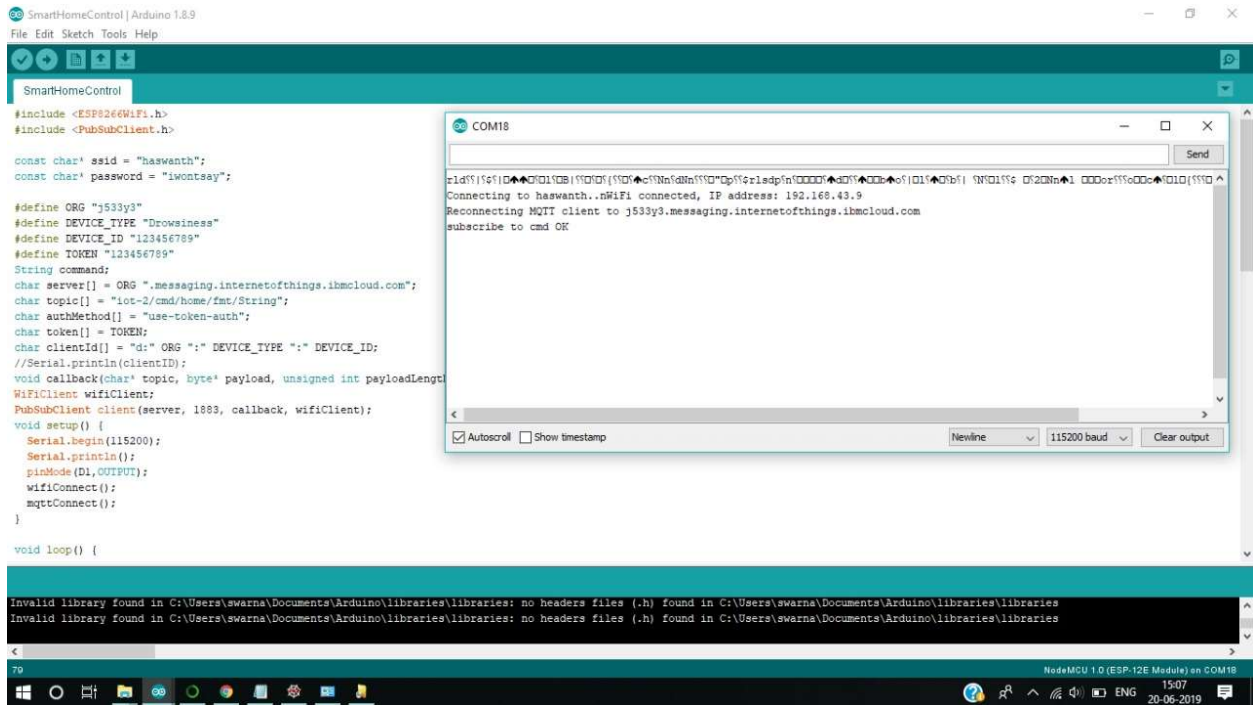
#define ORG "j533y3"
#define DEVICE_TYPE "Drowsiness"
#define DEVICE_ID "123456789"
#define TOKEN "123456789"
String command;
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char topic[] = "iot-2/cmd/home/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
//Serial.println(clientId);
void callback(char* topic, byte* payload, unsigned int payloadLength);
WiFiClient wifiClient;
PubSubClient client(server, 1883, callback, wifiClient);

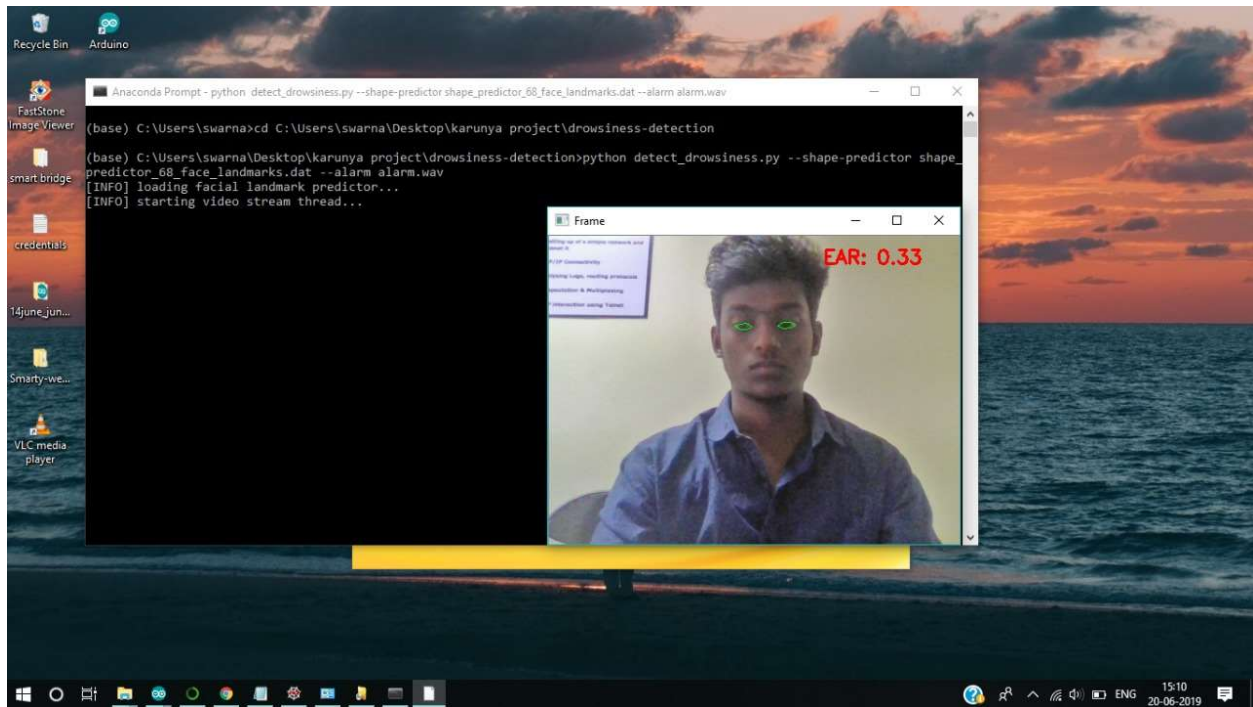
void setup() {
  Serial.begin(115200);
  Serial.println();
  pinMode(D1, OUTPUT);
  wifiConnect();
  mqttConnect();
}

void loop() {

Done uploading.
Invalid library found in C:\Users\swarna\Documents\Arduino\libraries\libraries: no headers files (.h) found in C:\Users\swarna\Documents\Arduino\libraries\libraries
Invalid library found in C:\Users\swarna\Documents\Arduino\libraries\libraries: no headers files (.h) found in C:\Users\swarna\Documents\Arduino\libraries\libraries

79 NodeMCU 1.0 (ESP-12E Module) on COM18
15:07
20-06-2019
```





## **5.CONCLUSION**

This task speaks to a case of methodical way to deal with the appraisal of wearable sensors for physiological parameter estimation. If it is settled in vehicles and utilized as an open source. By this undertaking driver's laziness is checked persistently. In a bad position, they will be frightened and display effortlessly. Of late according to IRB overview it had shown that 27 % street mishaps are expected the driver weakness, tiredness and laziness. In order to keep away from street mishaps outlining driver sluggishness location framework will be absolutely productive. In this way it is presumed that eye checking framework is one of the quick technique for influencing driver to alarm if sluggishness or exhaustion is experienced while driving. It is a standout amongst the most proficient strategy that can help us to lessen occurrences of lethal street mischances caused because of the driver weakness, languor and tiredness. We built up a framework that restricted and track the eyes and head developments of the driver keeping in mind the end goal to recognize laziness. The framework utilizes a blend of layout based coordinating so as to restrict the eyes. Amid following, framework will have the capacity to choose if the eyes are open or shut and whether the driver is looking in front. At the point when the will be shut for a really long time, a notice flag will be given as bell or caution pack message. In future, this venture will be improved in flying machine framework for quick and precise execution.

## **6.FUTURE SCOPE**

This application is basically written as a solution to the drawbacks of existing system. This application can be used as a real-world application and by any organization. We can import the code into any device, and we can use sensors like alcoholic sensor, smoke sensor to detect the driver's state of action. By this way we can protect the person from dangerous state. We can make this system more automated by implementing the automation of car when any alert comes due to the drowsiness of driver, car can automatically slow down to a safe side of the road and stop for a while and make the driver safe. This would be the best real-world application can be implanted in future.

## 7.REFERENCES

- [1] <https://www.pyimagesearch.com/2017/05/08/drowsiness-detection-opencv/>.
- [2] [https://en.wikipedia.org/wiki/Driver\\_drowsiness\\_detection](https://en.wikipedia.org/wiki/Driver_drowsiness_detection).
- [3] <https://web.stanford.edu/class/cs231m/projects/final-report-raghavan.pdf>.
- [4] <https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/>.
- [5] <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.69.3667&rep=rep1&type=pdf>.
- [6] Arun Sahayadhas , Kenneth Sundaraj and Murugappan Murugappan, Detecting Driver Drowsiness Based on Sensors: A Review,Sensors2012,12,16937-16953; doi:10.3390/s121216937, ISSN 1424-8220.