Driver's Drowsiness Detection and Alerting with Emergencyauto Parking System Based on IoT

[1]Kadimisetty Madhav,[2]Garapati Anjali Devi,[3]Sankula Kalyan Sai, [4]Koppineedi Madhu Kumar; [5] G.V.Vinod

[1,2,3,]UG Students, Department of Electronics and Communication Engineering, Godavari Institute of Engineering and Technology (A), Rajahmundry, Andhra Pradesh, India.

[4,]Assistant Professor, Department of Electronics and Communication Engineering, Godavari Institute of Engineering and Technology (A), Rajahmundry, Andhra Pradesh, India

Abstract: Drowsiness is that the main reason for the road accidents that are happening. Drowsiness may be a process during which one level of consciousness is reduced thanks to lacking of sleep or exhaustion and it's going to cause the driving force fall under sleep quietly. Various studies have suggested that around 20% of all road accidents are due to fatigue-related only, up to 50% on certain usual roads. quite 25% of highway traffic accidents that are caused as a results of drivers drowsiness. to scale back the danger of an accident by warning the driving force of his/her drowsiness we've developed this project. during this project we detect the driving force drowsiness and alert the driving force with buzzer, albeit the driving force doesn't get alerted the system automatically parks the vehicle and it'll send the situation coordinates to the relations . This project is especially supported the subsequent process 1) Face and Eye detection: Performs scale invariant detection using Haar Cascade Classifier perform through a webcam. 2) Eye feature extraction: Eye features are extracted using Hough Circle and 3) Extract single eye 4) Euclidean Distance calculation with threshold and perform drowsiness detection thereon .

5)Vehicle speed control 6) Parking and sending location. within the proposed method, following the face detection step, the facial components those are more important and thought of because the best for drowsiness, are extracted and tracked in video sequence frames. The contribution work is when drowsiness detected, after it'll give alarm to the driving force using buzzer if the driving force doesn't respond it waits for the edge limit after crossing the edge limit the system automatically parks the vehicle by giving the signal indication and after parking it send the present location coordinates to the relations

KEYWORDS - Drowsiness, HaarCascade Classifier, Hough Circle, Image Processing, Real Time Drowsiness Detection, OpenCV, 68 Landmark co-ordinates, L293 Motor driver,

DOI of the Article: https://doi.org/10.46501/GIETEC04

Available online at: https://ijmtst.com/icetee2021.html

As per UGC guidelines an electronic bar code is provided to seure your paper

To Cite this Article: Kadimisetty Madhav; Garapati Anjali Devi; Sankula Kalyan Sai; Koppineedi Madhu Kumar and G.V.Vinod. Driver's Drowsiness Detection and Alerting with Emergencyauto Parking System Based on IoT. *International Journal for Modern Trends in Science and Technology* 2021, *7*, pp. 18-26. https://doi.org/10.46501/GIETEC04

Article Info. Received: 18 May 2021; Accepted: 25 June 2021; Published: 30 June 2021

INTRODUCTION

Drowsiness and Fatigueness are often used synonymously in the driving state description of a driver. Involving multiple human factors, it's multidimensional in nature that researchers have found difficult to define over past decades. Despite the anomaly surrounding fatigue, it's a critical factor for driving safety. Studies have shown that fatigue is one among the leading contributing factors in traffic accidents worldwide. it's particularly critical for occupational drivers, like drivers of buses and heavy trucks, thanks to the very fact that they'll need to beat a protracted duration of the driving task, during the height drowsiness periods (i.e., 2:00 A.M. to 6:00 A.M. and 2:00 P.M. to 4:00 P.M.), and under heavy working conditions. Drowsy driving is becoming one among the foremost important explanation for road accidents. consistent with many surveys around 20% of road accidents is thanks to the driving force fatigue and therefore the percentage is increasing per annum. Drowsiness are often thanks to the adverse driving conditions, heavy traffic, workloads, late night long drive etc. Lack of sleep, absence of rest, taking medicines also are causes for drowsiness. When driver drives for quite the traditional period fatigue is caused and therefore the driver may feel tiredness which can cause driver to sleepy condition and loss of consciousness, this results in many accidents and death of the drivers. Drowsiness may be a phenomenon which is that the transition period from the awake state to the sleepy state and causes decrease in alerts and conscious levels of driver. it's difficult to live the drowsiness level directly but there are many indirect methods to detect the driving force fatigue. during this the most techniques used for blink detection is Eye ratio (EAR) method. The Ear method is completed by calculating a quantity named EAR. In normal condition the worth of EAR is nearly constant. If the driving force is in fatigue the EAR value are going to be approximately almost zero. Thus we will detect whether the driving force is in fatigue or not. Eye ratio (EAR), which is calculated by the coordinates of landmarks, which is employed to spot the state of eyes (open or closed). Thus if the eyes are open Eye ratio (EAR) is constant and reads a worth greater than zero, but when the eyes are closed then is Eye ratio (EAR) is on the brink of zero, which indicates the driving force is in fatigue condition.

This paper explains about the driving force drowsiness detection and automatic vehicle parking. it's a picture processing technique which is completed by OpenCv library and Dlib. during this paper we propose a way to extend drowsiness detection efficiency, merging the attention closure with the assistance of Dlib Frontal Face Algorithm which ends up in additional accurate data. The proposed method is predicated on the countenance of the driving force captured by a camera installed ahead of the driving force We used pwm motor controllers to demonstrate the vehicle hamper procedure and parking system.The vehicle gets hamper when it receives the drowsiness alert and therefore therefore the vehicle gives parking signal indication given by the signal lights and the vehicle gets parked. After Parking the situation co-ordinates are grabbed by GPS module and therefore the location coordinates are sent by the GSM module to the relation.

RELATED WORK (OR)LITREATURE REVIEW

2.1 Webcam

OpenCV may be a Library which is employed to hold out image processing using programming languages like python. This project utilizes OpenCV Library to form a Real-Time Face Detection using webcam as a primary camera.

2.2 Image Resize

Acquisition of image takes place. It is used to resize input image into standard image format.open cv haar cascade classifier supports either a single face detection or multiple faces detection (i.e.,thatspecialize in the people during a picture). We will easily create a face thumbnail of users, crop an image to specializein an individual or resize an image to fill the required dimensions while making sure the person in the original picture appears within the resized version of the image.

2.3 Haar cascade classifier

Object Detection using Haar feature-based cascade classifiers is an efficient object detection method proposed by Paul Viola and Michael Jones in their paper, "Rapid Object Detection employing a Boosted Cascade of straightforward Features" in 2001. it's a machine learning based approach where a cascade function is trained from tons of positive and negative

images. it's then wont to detect objects in other images. Here we'll work with face detection. Initially, the algorithm needs tons of positive images (images of faces) and negative images (images without faces) to coach the classifier. Then we'd like to extract features from it. For this, Haar features shown within the below image are used. they're like convolutional kernel. Each feature may be a single value obtained by subtracting sum of pixels under the white rectangle from sum of pixels under the black rectangle. Now, all possible sizes and locations of every kernel are wont to calculate many features. for every feature calculation, we'd like to seek out the sum of the pixels under white and black rectangles. to unravel this, introduced the integral image. However large a picture, it reduces the calculations for a given pixel to an operation involving just four pixels. It makes things superfast.

Haar-cascade Detection in OpenCV

OpenCV comes with a trainer also as detector. Its function travel by Training. Here we'll affect detection. OpenCV contains many kind of pre-trained classifiers for the following like eyes,face, smiles, etc. Those XML files are stored within the OpenCV/data/haarcascades/ folder. At first It creates a face and eye detector with the help of OpenCV. First we'd like to load all the specified XML classifiers that are required. Then we will be loading our input image (or video) in the grayscale mode.

```
Import numpy as np
import cv2 as cv
face_cascade=cv.CascadeClassifier('haarcascade_frontal
fac
e_default.xml')
eye_cascade =
cv.CascadeClassifier('haarcascade_eye.xml')
img= cv.imread('sa.jpg')
gray = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
```

Now we discover the faces within the image. when founds the faces, then it returns the positions of detected faces as in the specified syntax format Rect(x,y,w,h). Once we get these locations, we will create a ROI for the face and apply eye detection on this ROI ,since eyes are always on the face.

2.4 Dlib face detector

While the library is originally written in C++, it's good, easy to use Python bindings. it's used dlib for face detection and facial landmark detection. The frontal face detector in dlib works very well. it's simple and just works out of the box. This detector is based on histogram of oriented gradients (HOG) and linear SVM. While the HOG+SVM based face detector has been around for a brief time and has gathered an honest number of users. The CNN (Convolutional Neural Network) based face detector libraries are made available in dlib. The HOG based face detector in dlib, could also be an honest "frontal" face detector and it's indeed.

Dlib Facial Landmark Detector

We use dlib and OpenCV to detect facial landmarks during an picture. Facial landmarks are implemented to localize and represent salient regions of the face, such as:

1.Eyes

2. Eyebrows

3. Nose

4. Mouth

5. Jawline

Facial landmarks are successfully applied to face alignment, head pose estimation, face swapping, blink detection and far more. The fundamentals of facial landmarks, includes:

1. Exactly what facial landmarks are and the way they work.

2. The way to detect and extract facial landmarks from a picture using dlib, OpenCV, and Python.

The pre-trained facial landmark detector inside the dlib library is used to estimate the location of 68 (x, y)-coordinates that map to facial structures on the These annotations are part of the shape predictor 68.dat file which the dlib facial landmark predictor was trained on. Regardless of which dataset is used, the same dlib framework can be leveraged to train a shape predictor on the input training data — this is useful if we would train facial landmark detectors or custom shape predictors.

Detection ofFacial landmarks with dlib, OpenCV, and Python

The first utility function is rect_to_bb , short for "rectangle to bounding box": This function accepts one argument, rect , which is assumed to be a bounding box rectangle produced by a dlib detector (i.e., the face detector). The rect object contains the (x, y)-coordinates of the detection. However, in OpenCV, we normally consider a bounding box up terms of "(x, y, width, height)" so as a matter of convenience, the rect_to_bb function takes this rect_object and transforms it into a 4-tuple of coordinates. The dlib face landmark detector which can return the form of the thing which containing the 68 (x, y)-coordinates of the facial landmark. Using the shape_to_np function, we will convert this object to a NumPy array. We used these two helper functions, to detect facial landmarks in images.

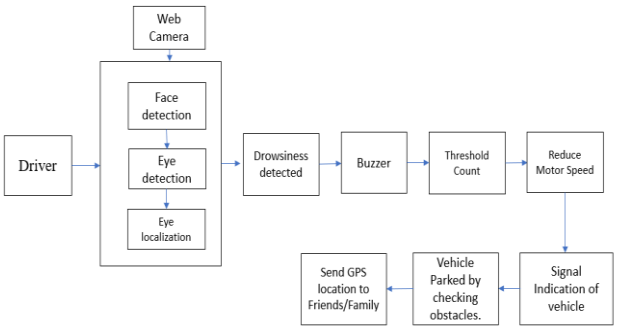
2.5 Eye Aspect Ratio & Alert Buzzer

Duringthis system , we are using different landmarks to detect the opening and shutting of eye. This landmark detector that capture most of the characteristic points on a person's face image. the eye blink could also be a quick closing and reopening of an individual's eye. Each individual person features a touch different pattern of blinks. The pattern differs within the speed of closing and opening of the eye, a degree of compressing the eye and through a blink duration. the eye blink lasts approximately 100-400ms. From the landmarks detected within the image, we derive the eye ratio (EAR) that's used as an estimate of the eye openingstate. for every video frame, the eye landmarks are detected. the eye ratio between height and width of the eye is computed.P1,P2,…,P6 are the landmarks on the eye .It is used to compute the ratio of distances between the vertical eye landmarks and thus the distances between the horizontal eye landmarks. The return value of the eye ratio are getting to be approximately constant when the eye is open. the price will then rapidly decrease towards zero during a blink.

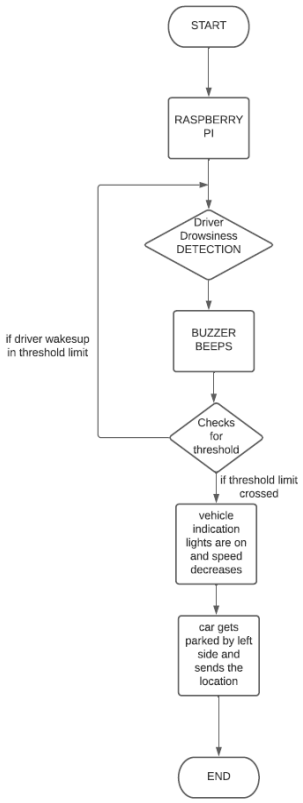
Proposed System

To improve the accuracy as well as to reduce the execution time of fatigue, drowsiness detection system and Vehicle automatic parking and location sending would be done as shown in Block diagram and Flowchart :

Block Diagram:



Flowchart:



3.1. Webcam

We use webcam to monitor the driver's face which is connected to raspberry pi.

3.2. Image Resize

It is used to resize input image into standard image format.

3.3. Haar Cascade Classifier

A Haar cascade classifier is an algorithm which is employed to detect the thing which it has been trained

forfrom the source. The haar cascade is by superimposing the positive image over a group of negative images. The training is mainly done on a server and goes on various stages.

3.4. Dlib face detector

It is used to find and locate the face within the image. It initializes dlibs pre-trained face detector based on a modification to the standard histogram of oriented gradients (HOG).

3.5. Facial Landmark 68 R.O.I

The pretrained facial landmark detector inside the dlib library is used to estimate the situation of 68 (x , y)-coordinates face that map to facial structures of the face. These annotations are a neighborhood of 68 point shape predictor 68.dat which the dlib facial landmark predictor was trained on. The facial landmark detection is used to localize each of the important regions of the face.

3.6. Eye Region of Interest

Extracting exact eye locations takes place. Image cropping is employed to limit the planet of labor nearer to the eyes since the activity of the eyes we consider. the whole area of the image is reduced by cropping the image to 2 fifth to 3 fifth of the whole area of the image on the upper region with the result that the separation of eyes is performed.

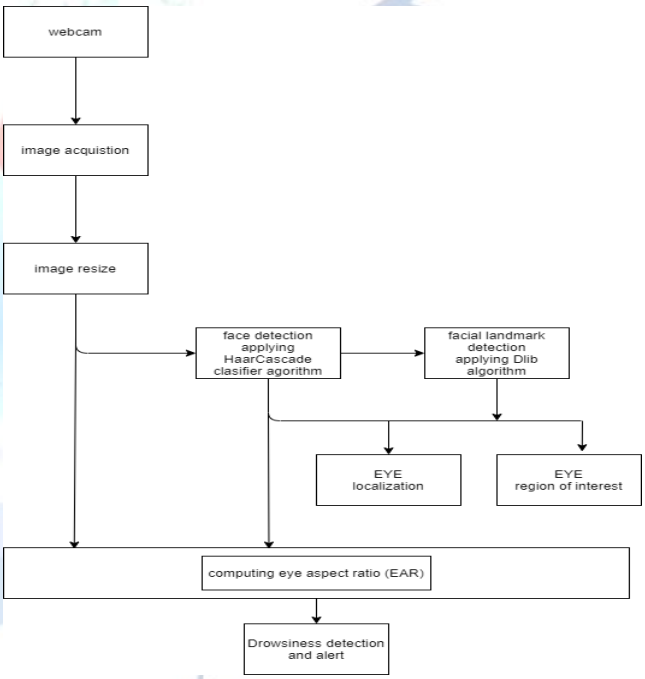
3.7. Eye localization and drowsiness detection

It is used to compute the ratio of distances between the vertical eye landmarks and thus the distances between the horizontal eye landmarks. The return value of the eye ratio are getting to be approximately constant when the eye is open. the price will then rapidly decrease towards zero during a blink. the eye blink could also be a quick closing and reopening of an individual's eye. Each individual person features a touch different pattern of blinks. The pattern differs within the speed of closing and opening of the eye , a degree of compressing the eye and through a blink duration. the eye blink lasts approximately 100- 400ms. From the landmarks detected within the image, we derive the eye ratio (EAR) that's used as an estimate of the eye opening state. for every video frame, the eye landmarks are detected. the eye ratio between height and width of the eye is computed. From the fig. 2 P1,P2,…,P6 are the landmarks on the eye.

$$\underline{||P2-P6|| + ||P3-P5||}$$

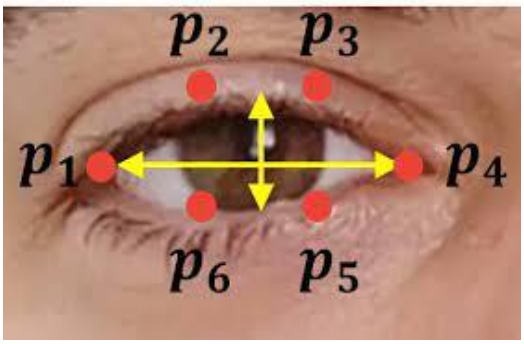
$$EAR = 2||P1 - P4||$$

where P1,…..,P6 are the 2D landmark locations on the eye . The EAR is typically constant when an eye fixed is open and is aged the brink of zero while closing an eye. Since eye blinking is performed by both eyes synchronously, the EAR of both eyes are taken and it's averaged. After getting the EAR value, if the price may be a smaller amount than the limit for 2 or 3 seconds the drive is claimed to be drowsy. After detecting fatigueness of the driving force , drowsiness with EAR threshold value runs and therefore the alert alarm runs then it shows the message as “Drowsiness Detected” within the interface.



3.8. Eye Aspect Ratio(EAR)

Duringthis system, we are using different landmarks to detect the opening and shutting the attention. This landmark detector that captures most of the characteristic points on an individual's face image. it's wont to compute the ratio of distances between the vertical eye landmarks and therefore the distances between the horizontal eye landmarks. The return value of the attention ratio is going to be approximately constant when the attention of the eye is open. the worth will then rapidly decreasestowards the zero during a blink.



3.9 PWM CONTROLLER

We use l298n PWM(pulse width modulation) based motor controller for the speed control of the vehicle. once we receive the signal from the raspberry pi that the driving force is in fatigue, then the Arduino nano controller receives the signal then it process the pwm commands in such how that the speed of the motors get decreased gradually.

3.9.1 Automatic Parking

When the signal is recieved from the raspberry pi, the system checks for any vehicles coming from the left/right side. If there are no vehicle coming from the side it signals by blinking the left side signal lights and then it gradually comes to left. And the car gets parked at the left side. We use ultrasonic sensors to check whether any vehicles coming from the side.

3.9.2 GPS location

We send the gps location of the car after the car has been parked with the help of the gsm module. We collect the GPS co-ordinates from the gps module and send them to the relatives/friends through the gsm module.

HARDWARE USED

4.1Sim 900A Gsm Module:

SIM900 GSM/GPRS shield also can be termed as GSM modem, which may be integrated into a bigger number of IoT projects.Wewill use this shield to accomplish almost anything a normal cell phone can: SMS text messages, Make or receive phone calls, connecting to internet through GPRS, TCP/IP, and more. To top it off, the shield supports quad-band GSM/GPRS network, meaning it works pretty much anywhere in the world.

The SIM900 shield packs a surprising amount of features into its little frame. Some of them are listed below:

- Supports: GSM850, EGSM900, DCS1800 and PCS1900
- Connect onto any global GSM network with any 2G SIM network.
- Make and receive voice calls using an external earphone & electret microphone
- Send and receive SMS messages
- Send and receive GPRS data (TCP/IP, HTTP, etc.)
- Serial-based AT Command Set
- U.FL and SMA connectors for cell antenna
- Accepts Full-size SIM Card

LED Status Indicators:

There are three LEDs fixed on the SIM900 GSM/GPRS shield which indicates connectivity status. With the assistance of those LEDs you'll get a visible feedback on what's happening on with the shield.

PWR: This LED is connected to the shield's power supply line . If this LED is in ON condition, then the shield is receiving power.
Status: This LED indicates SIM900's working status. If this LED is on, the chip is in working mode.
Netlight: This LED indicates the status of your cellular network. It'll blink at various rates to point out which state it's running on.

- off: This condition tells that the SIM900 chip isn't running.
- 64ms is on, 800ms is off: The SIM900 chip is running but not registered to the cellular network yet.
- 64ms is on, 3 seconds is off: The SIM900 chip is registered to the cellular network & can send/receive voice and SMS.
- 64ms is on, 300ms is off: The GPRS data connection that you simply had requested is active.

In our project we used this GSM module to send the location co-ordinates details to the predefined numbers.

4.2 GPS Module:

It consists of internal RTC battery and can be directly connected to USART of the microcontroller. This module gives the precise location within the format of latitude, longitude and also we get time also. GND is that the Ground Pin and wishes to be connected to GND pin on the Arduino. TxD (Transmitter) pin is employed

for serial communication and.RxD (Receiver) pin is employed for serial communication. VCC supplies power for the module. we will directly connect it to the 5V pin on the Arduino. But we connect it over the facility supply pins. GPS data is displayed in various message formats over a serial interface. There are both standard and non-standard message formats. Nearly all GPS receivers output NMEA data. The NMEA standard is formatted in lines of data called sentences. Each sentence contains various bits of knowledge employed in comma delimited format. Once a GPS module is powered, NMEA data (or another message format) is shipped out of a serial transmit pin (TX) at an specific baud and update rate, albeit there's no lock. to possess our microcontroller read the NMEA data, all that's needed is to attach the TX pin of the GPS to the RX (receive) pin on the microcontroller. To configure the GPS module, we'll got to also connect the RX pin of the GPS to the TX pin of the microcontroller.

4.3 L298N Module:

L298NMotor Driver is interfaced with Arduino to regulate the motors. it'll control both the speed and spinning direction of the DC motors. so as to possess an entire control over DC motors, we've to regulate its speed and rotation direction. this will be done by combining these two techniques.

- PWM – For controlling speed
- H-Bridge – For controlling rotation direction

The speed of a DC motor are often controlled by varying its input voltage i.e., a typical technique for doing this is often to use PWM (Pulse Width Modulation). it's a way where average value of the input voltage is controlled by sending a series of ON-OFF pulses, then the standard voltage is proportional to the width of the pulses referred as Duty Cycle. the upper the duty cycle, the greater the typical voltage being applied to the dc motor(High Speed) in order that the motor rotates with high speed or the very best rpm it's specified and therefore the lower the duty cycle, the less the typical voltage being applied to the dc motor(Low Speed) in order that the motor rotates with low speed or lowest rpm it's specified. We used these PWM pins for controlling the speed of the vehicle when the Driver's Drowsiness is detected. With the assistance of those pwm pins we reduce the speed gradually and parks the vehicle.

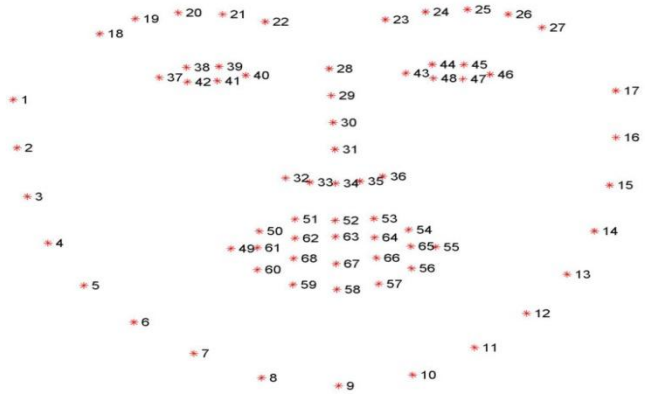
4.4 Ultrasonic Sensor:

The HC-SR04 Ultrasonic sensor has two ultrasonic transducers. The one acts as an transmitter converts electrical signal into 40KHz ultrasonic sound pulses. The receiver listens for the transmitted pulses obtained above. If it receives, it'll produce an output pulse whose width is wont to determine the space that the heart beat travelled. The sensor is little and hence it offers excellent non-contact range detection between 2 cm to 400 cm (about an in. to 13 feet) with an accuracy of 3mm. Since it operates on 5 volts, it are often connected on to an Arduino board. The ultrasonic sensors are attached to the car chasis in such how that the side coming vehicles and other vehicles are monitored by calculating the space between the 2 vehicles. VCC provides the facility supply for HC-SR04 Ultrasonic distance sensor which we connect the 5V pin on the Arduino board. Trig pin is employed for triggering the ultrasonic sound pulses. Echo pin modulates a pulse when the reflected signal is received back. The length is proportional to the time it takes for the transmitted signal to be detected. GND should be coupled to the bottom of Arduino. When a pulse of a minimum of 10 μS in duration is given to the Trig pin. Then the sensor transmits a sonic burst of eight pulses at 40KHz. This 8-pulse pattern makes the ultrasonic sensor unique from other devices, allowing the receiver to match the transmitted pattern from the ambient ultrasonic noise. The Echo pin goes HIGH to start out the start of formation of the echo-back signal. If the pulses aren't reflected some time past the Echo signal are going to be gone after 38mS and return low. Thus a 38mS pulse shows that there's nothing wrong within the range of the sensor. If the pulses are reflected back the Echo pin goes when the signal is received. this may provide a pulse varies between 150 μS to 25mS, counting on the time it took for the signal to be received. The width of the received pulse is then used for calculating the space to the reflected object. this may be calculated by using the straightforward distance-speed-time equation.

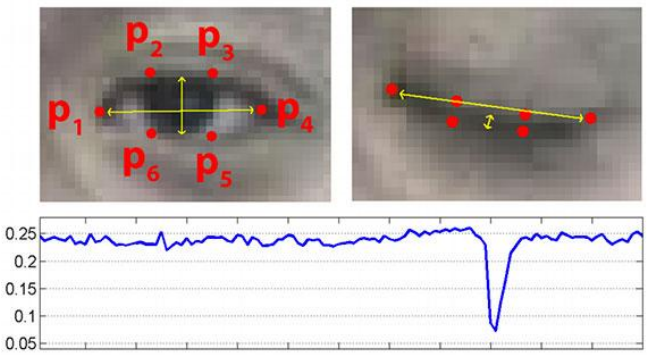
RESULTS AND DISCUSSION

Facial Landmark detection and identification of eye coordinates for face detection takes place. In this a dlib face detector used to locate a face in an image and dlib facial landmark predictor used to locate landmark.

5.1 Eye Aspect Ratio (EAR) Output



dlib's 68 facial landmarks are indexable which enables us to urge the varied facial expressions using Python array slices. Given the facial landmarks associated with an eye fixed, we will apply the attention ratio (EAR) algorithm.

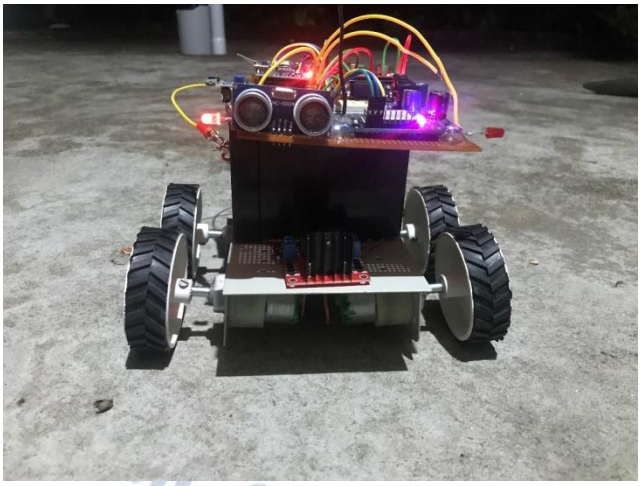


On the top-left we've fixed an eye fixed that's open and therefore the eye facial landmarks has been plotted. Then on the top-right we'll consider for an eye fixed that's closed. rock bottom plots shows the attention ratio over time. As we observed, the attention ratio is constant (indicating that the attention is open), then rapidly drops to shut to zero, then raised again, indicating that a blink occurred.

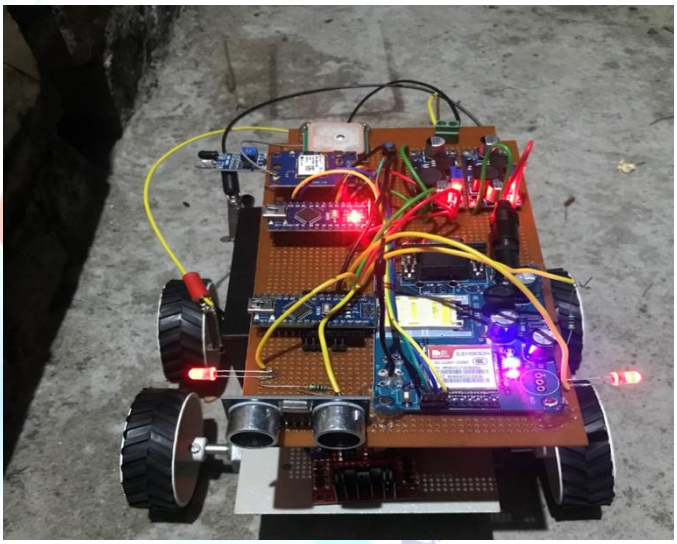
5.2 Graphical Analysis

In graphical analysis, we'll consider EAR on y-axis and Time(sec) on x-axis. The EAR measurements plots with regard to time. Each dip in EAR curve corresponds to blink and width of EAR dip represents duration of eye closed and formed basis of detection. Algorithm performs well in repeated experiments on different sort of faces. EAR threshold adjustment provided in GUI front is beneficial for fine tuning of algorithm on different sort of faces. No false detection and missed detection of blinks and closed eye for a few period was observed.

5.3Vehicle Parking with signal indication:

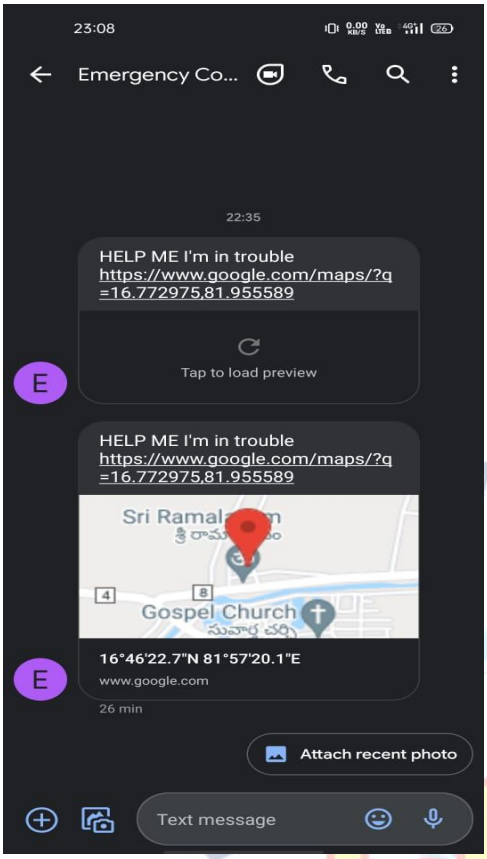


5.4After parking the Vehicle :

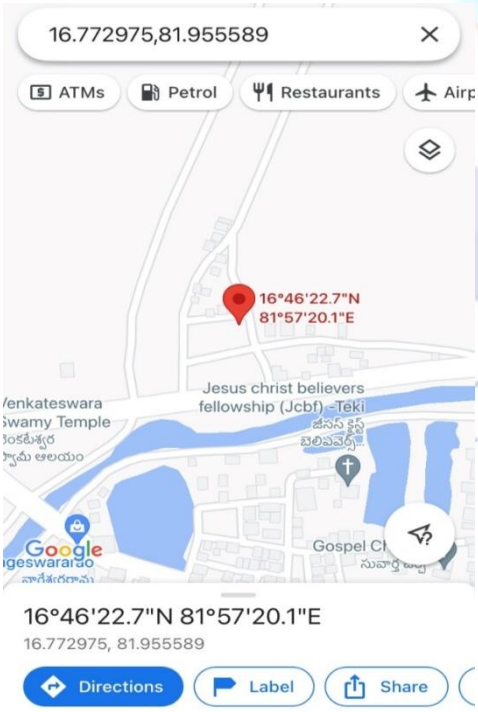


After the vehicle gets parked by the road side both the signal lights are indicated.

5.5 Sending Location through GSM:



5.6GPS LOCATION COORDINATES :



CONCLUSION

Driver'sDrowsinesss detection and automatic vehicle parking system was developed in order to reduce the count of accidents that are happening. This system can reduce the risk of happening accident by detecting the drowsiness of the driver with the help of the webcam attached to the raspberry pi module, this detection process involves webcam, haar cascade classifier, facial landmark detection are used to calculate whether or not a driver is drowsy. After the driver drowsiness is detected the pwm motor controllers control the speed and turns on the indicator lights and gives the signal to park side and the vehicle gets parked. Then the location coordinates are sent from the gps and gsm modules.We used the PWM pins for controlling the speed of the vehicle when the driver's drowsiness is detected. With the help of these pwm pins we reduce the speed gradually and parks the vehicle.The ultrasonic sensors are attached to the car chasis in such a way that the side coming vehicles and other vehicles are monitored by calculating the distance between the two vehicles.

REFERENCES

1. Open Source Computer Vision Library Reference Manual
2. Facial landmarks with dlib, OpenCV and Python: https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/
3. Real-Time Eye Blink Detection using Facial Landmarks: http://vision.fe.uni-lj.si/cvww2016/proceedings/papers/05.pdf
4. https://www.arduino.cc/reference/en/
5. https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/

