Group 8: SAFE Vehicle System using IoT

Gajender Sharma Kajal Sethi Pranshu Sahijwani
Shivam Tripathi
21111028, 21111053, 21111048, 21111408
{gajenders21, kajals21, pranshus21, shivamtr21}@iitk.ac.in
Indian Institute of Technology Kanpur (IIT Kanpur)

Abstract In recent years, the casualties of traffic accidents caused by drivers continue to increase mainly due to their negligence in driving or obeying proper safety measures while driving. Development in technologies in IoT and Artificial Intelligence allows us to make a vehicle packed with many functionalities which makes it less accident prone by assisting and alerting the driver about safety checks and other unusual activities about driver like driver drowsiness, a distraction from the mobile phone.Our project has various functionalities like Drowsiness detection, driver temperature monitoring, face mask detection which is crucial in times of COVID-19 and seat belt fasten detection, alerting the driver if any detector detects the negligence, and also notifying the family members and rescue authorities if some mishap has happened despite all the safety measures with location and intensity of the accident.

1 Introduction

Cultures all across the world are concerned about safe driving. So we have tried to build a model that would rescue us from potential accident and make our travel safer. We have looked for signs of fatigue, a mask (in some areas, wearing a mask inside a vehicle is required due to poor air quality as well as COVID guidelines), and a seat belt. A car accident is caused by a combination of factors including vehicle malfunction and human activity.
Multiple surveys have been performed in order to find the frequent cause of road accidents and according to a survey conducted in India [1], there are a variety of reasons for road accidents, the most common of which is a distracted driver, which includes common behaviors such as using a cell phone, weariness, daydreaming, taking calls, and so on. Similar facts are also stated by another survey which was conducted by DSC Attorneys in South Africa [6].
Except for driver tiredness, most of these problems are daily fixes. Many tired drivers are unaware they are sleeping, even briefly and accidents can happen at any time and anywhere. Lack of rescue has also resulted in severe damage. A report or an emergency call was not made due to the event's unpredictable nature.Thus underscoring the reality that much more can be done to make travel safer in cars. If the driver does not wake up within a particular time frame, the system should alert family members and authorities. Rescuers can save lives if they act promptly. So we created our safe vehicle with all of this in mind.

2 Related Work

Several steps can be taken to make a vehicle safer for the driver, including recognising seat belts, masks, tiredness of the driver, and in case of accident, sending notifications to the concerned authorities and family members. As detailed in the following sections, drowsiness of a driver can be identified in a variety of ways employing vehicle-based, psychological, and behavioural metrics performed through various predictive algorithms [12].

2.1 Implementation of Detection System for Drowsy Driving Prevention Using Image Recognition and IoT [16]

The techniques that use cameras to detect drowsiness have a very low accuracy and do not work well in practice. The study presented in this paper uses facial and eye-blink recognition based on image data along with data from a CO2 concentration sensor. The drowsiness detection system implemented in the paper consists of three main parts:

- The image data is collected from the camera and cleaned using image processing techniques. The region-of-interest is extracted using Canny edge detection and eye-pupil are detected using Hough Circle Transform. In the extracted region, machine learning techniques are applied which improves the detection performance.
- The CO2 concentration detection sensor supplies crucial data to detect possible drowsiness. The presence of high CO2 content in vehicles is one of the leading factors that can increase drowsiness and can even be fatal to driver health.
- The whole design is implemented in a box that is mounted in front of the driver. If the drowsiness is detected, the box automatically ventilates the interior of the car and plays the music to alert the driver.

2.2 Smart Detection and Prevention system for Drowsy Driving[7]

The current framework, for example, vehicle-based and physiological estimations(Example: ECG, EEG etc), are intrusive and it has numerous disadvantages. They may provide better accuracy in detecting the drowsiness, but also tend to distract the driver. Our proposed framework is non-intrusive and profoundly powerful in estimating and avoidance of lazy driving. In this procedure, the Raspberry-pi is utilized to get the facial landmarks and to distinguish the EAR(Eye Aspect Ratio) value of the individual.

2.3 Car Crash Prevention And Detection System[15]

Vibrator sensor, which on detection of the crash will activate the circuit and an S.O.S message will be sent to the nearest hospital, police station, fire brigade and emergency number of the user. There are blind curves in the hilly areas and it is highly difficult to detect if there is a car on the other side of the road, especially in the foggy or misty weather. Smart poles will be used to notify the drivers on each side.These poles use a transmitter and receiver mechanism and alert the driver on the other side using a sharp red light or an alarm.

2.4 Raspberry PI controlled SMS-Update-Notification (Sun) system[19]

It implements a notification system in which Short messages service (SMS) can be sent through mobile using the IoT GSM module for displaying the notification and a website is maintained to show the notification. To send and receive messages a SIM card is used in the GSM module of the IoT device that is connected to the Raspberry PI and a database like MySQLdb is also used to store and organize notifications.

2.5 IoT-Based Smart Alert System for Drowsy Driver Detection [12]

The paper addresses a drowsy driver alert system that has been developed in which the Video Stream Processing (VSP) is analyzed by eye blink concept through an EAR (threshold 0.25) and Euclidean distance of the eye. Face landmark algorithm is also used as a proper way to eye detection as facial landmark like monitors the driver facial behaviour, as they doesn't change under any ordinary situation thus it is more accurate than the cascaded method, also these characteristics help to measure driver fatigue and instantly alert him with the help of voice speaker.

2.6 Seat belt fastness detection based on image analysis from vehicle in-cabin camera[17]

Seat belt fastness is one of the critical safety features which keeps driver safe in case of potential accidents. Although many modern cars already occupied with seat belt detectors which can easily be tampered and tricked.There are many Machine learning algorithms which detects seat belt based on video feed but they are complex and not so efficient for automobile manufacturers to implement in their vehicles. This paper is based on YOLOv3(you only look once) object detection model for detecting seat belt fastness of the driver as opposed to many edge detection algorithms for seat belts because edge detection algorithms can be mostly inaccurate because there is high probability that they will detect any slope similar to seatbelt and inaccurately detect it.

2.7 Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV [13]

Sequential convolutional neural networks are used to recognise the face in a picture and determine whether a mask is worn. The approach is evaluated on two datasets, d1 [3] and d2 [4]. The dataset d1 contains 1376 photos, 690 of which feature persons wearing masks and the remainder of which do not. The d2 dataset has more photos with numerous persons in them. 853 images The algorithmic process begins by prepping the input picture collection for the CNN model. The image is grayscaled and resized. The pixel intensities are 0-1 normalised. Convolutional, ReLu, and max pooling layers make up the model. For classification loss, the Adam optimizer uses categorical cross entropy. The system was 95.77 percent accurate on dataset d1 and 94.58 percent accurate on dataset d2.
The approach detects partially obscured faces. If any of the face's areas are obscured, the model works correctly. It can use live data from surveillance cameras to recognise if someone is wearing a mask.

2.8 Vehicle speed control system using GSM/GPRS [9]

The project's main features were real-time tracking through SMS and real-time vehicle speed reduction. The team used GPRS for cellular data transmissions since it has excellent data transfer rates. GSM was chosen because it digitises and compresses. The Global Navigation Satellite System (GNSS) module was also employed.
After installation, the device uses GPS. This uses mobile data to track your whereabouts. One of the main functions of GPS is tracking. The satellite calculates the distance travelled before sending data to the receiver. The user's mobile device transmits location, speed, activity, fuel rate, and other data.

2.9 IOT—Eye Drowsiness Detection System by Using Intel Edison with GPS Navigation [11]

The paper focuses on real-time processing and driver fatigue detection. An LED indicator shows the system's mode. The smartphone camera on the windscreen captures video. The video is sent to the cloud for analysis. Simultaneously, the GPS navigation system tracks the driver's position. This data will be stored on the cloud. The cloud data will be sent to an Intel board for hardware communication. A GSM notification will be sent by the Intel CPU to the output system.
The system's performance was examined for several image resolutions, and the quickest was picked. Between 25 and 100 cm was found to be the optimal driver-camera distance. A green LED appears if the eyes are closed for 4 seconds. On for more than 4 seconds is the Yellow LED, Alarm, and LCD display 'Wake Up.' After 8 seconds, the red LED indicates the system is in 'Danger Mode' and sends a message to the designated contacts and emergency services.

3 Proposed Idea

Our goal is to create a real-time safety system that can detect driver drowsiness and alert the driver with a buzzer, check whether the seat belt is worn, and whether the mask is properly covering the face.

The project flow is as follows:

We check the driver's mask and his temperature. A buzzer alerts the driver if anything is amiss. We constantly monitor the driver. An alert is raised to wake up a driver who has fallen asleep or has closed his eyes for any reason. We monitor the driver and alert worried relatives if he exceeds a 10-second threshold. We monitor the driver and alert worried relatives if he exceeds a 10-second threshold. It can also detect a crash and alert the affected family members. It otherwise runs normally.

Our project has two major sections: training ML models to detect various safety aspects and simulation using sensors and code to control their activity based on the ML model predictions. As a full-featured development platform that takes you from concept to completion, we used Proteus 8 Professional. Innovative features such as automatic PCB wiring and layout are just a few of the advantages [14].

The algorithms that our team has worked on were developed for Face recognition, Eye gaze detection, Seat belt detection and Mask detection. The sensors/peripheral devices used for simulation of the project were: Raspberry pi 3 board, Vibration sensor(Vibration SW-420), GPS module, Temperature sensor(LM35), Compim to open serial communication, Relay module using a DC motor and a Push-Pull four channel driver with diodes(L293D), 16 x 2 LCD display(LM016L) and Buzzer module consisting of a buzzer(BUZ1) and a bipolar transistor(2N5550)

4 Methodology

4.1 Dataset

- **Eye gaze detection** -
  - **Synthetic Gaze and Face Segmentation:** This dataset contains high quality synthetic images of human heads along with facial landmark locations. [8]
  - **Unity Eyes:** This dataset contains synthetic images of eyes used to estimate the eye-gaze [5].
- **Seatbelt** - We have prepared a custom dataset by downloading the driver seatbelt images from google. The dataset contains around 315 images with each image depicting a driver with seat belt fastened from different angles. We have used it for our real time object detection model YOLOv3 in live video stream for seat belt detection.

4.2 Preprocessing

- For gaze detection we have calculated *Gaze angle* for each of the images from the complex JSON data corresponding to each image in the data-set. This angle was later used to calculate the accuracy. We used **arctan2** function provided in Python library - **numpy**.
- Seat belt Detection : we have created a custom dataset of seat belt images so that we can detect the seat belt of the driver using the yolov3 [18] object detection model for our custom dataset. So we have to label the images exactly the way our yolov3 detection algorithm [10] takes input for training the model. So we have labeled all the collected images with coordinates of the seatbelt in each image using an open-source tool called LabelImg.It created a text file corresponding to each image with four coordinates of the bounded box in the image we created with the top left and bottom right x and y coordinates.

- Face-Mask Detection: To detect the face-mask we have used two pre-trained models for face detection and mask detection respectively. The face-mask detection model is trained using mobilenetV2 which can be easily run on mobile devices. The network architecture for mask detection contains input layer followed by mobilenet (instead of conv-net), max-pooling, fully-connected, and output layer. We have used the pre-trained model for this task which is available on GitHub at [**?**]. The mask detection model uses the detected faces from the image. For face detection, a standard pre-trained resnet model is used which can detect multiple faces from the image.

4.3 Machine Learning Algorithms Used

- **Drowsiness Detection** - For detecting the eye drowsy status of the driver, we have to detect the status of eyes whether they are open or closed, The **dlib** library can be used to detect a face in an image and then find 68 facial landmarks on the detected face. Now we extract the eyes landmarks explicitly from these 68 coordinates for calculation of EAR(eye aspect ratio)(Figure 6.)

  EAR (eye aspect ratio) is a ratio of distances between two upper lid and lower lid coordinates divided by distance between horizontal coordinates of eyes. This ratio represent the state of the eyes whether they are wide awake or drowsy or closed based on the EAR ratio calculated. The standard threshold value of EAR for non drowsy eyes are 0.25 if the value comes out to be greater than 0.25 then driver is wide awake else he is drowsy or distracted from the road, if the ratio comes out to be 0 then eyes of the driver is closed.

$$EAR = \frac{|x_2 - x_6| + |x_3 - x_5|}{2 \times |x_1 - x_4|}$$

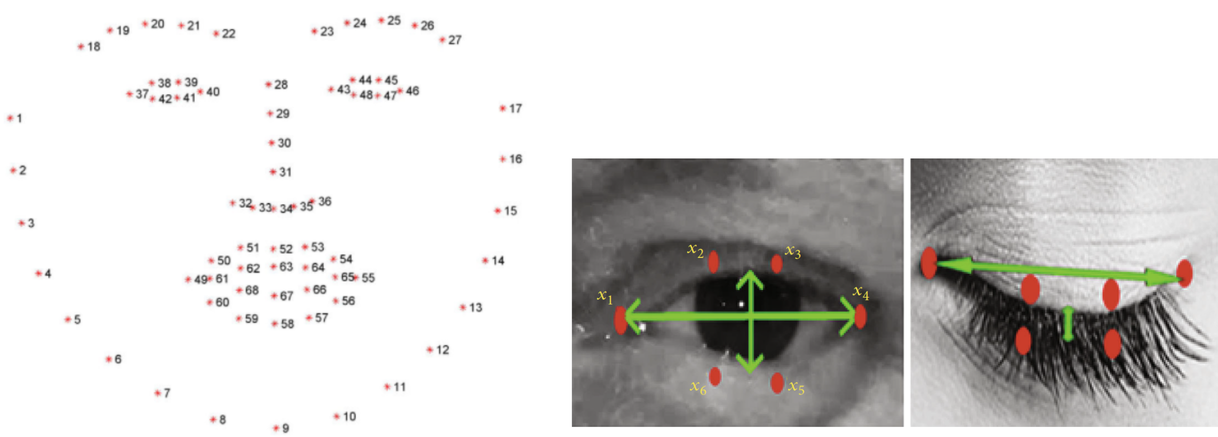

Figure 1: 68 facial(left) and 6 Eye(right) Landmarks for Eye blink detection[7]

- **Gaze Detection**:The approach was to train the EyeNet model using some data, and use to detect the Gaze. The **Eyenet Model**[2] has *Convolutional NNs* at its core. The model architecture is based on the stacked hourglass model[1] and was implemented using pytorch. We used 100k samples to train the model, and testing has been done using 15k samples. Accuracy: **95.1%**

- **Seat belt Detection**: Seat belt detection can be done with many edge detection ML algorithm but they are not accurate because it can mispredict various things slope similar to seatbelt as false seatbelt. So we are implementing real time object detection model YOLOv3(you only look once) which identifies objects in live feed, images, videos,etc. Yolov3 uses an underlying deep architecture called darknet-53 which has 53 layers trained on ImageNet,For the task of detection, 53 more layers are stacked onto it, giving us a 106 layer fully convolutional underlying architecture.YOLOv3 is a Convolutional Neural Network (CNN) for performing object detection in real-time. CNNs are classifier-based systems that can process input images as structured arrays of data and identify patterns between them (view image below). YOLO has the advantage of being much faster than other networks and still maintains accuracy.The YOLOv3 algorithm divides an image into a grid first. Each grid cell forecasts the placement of a certain number of boundary boxes around items that score well in the predefined classes.Each boundary box has a confidence score that indicates how accurate it believes the forecast should be, and each bounding box identifies only one object. To determine the most common forms and sizes, the boundary boxes are created by clustering the dimensions of the ground truth boxes from the original dataset.[10]

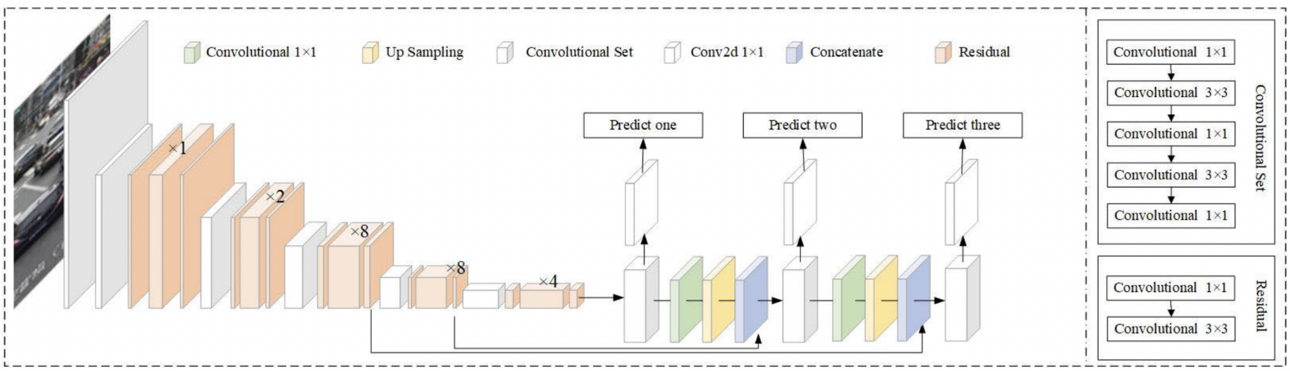

Figure 2: Architecture of YOLOv3 using underlying darknet-53[?]

4.4 Setup Required

We used a simulator to develop our project due to lack of physical interaction between team members and sensor and development board availability. Proteus 8 Professional is a full-featured development platform that takes you from the basic concept to the finish line. A few of the advantages include intelligent principle layout, hybrid circuit simulation and precise analysis, single-chip software debugging, PCB automatic layout and wiring [14]. Rest of the components are explained further.

- **Raspberry pi 3 board** - The key reason for choosing this board for the project was its extensive peripheral support and, most crucially, its compatibility for many programming languages such as C, C++, Python. It features 40 pins, including 4 power (two 5V and two 3.3V), 8 ground, and 28 GPIO (labelled 0 to 27), some of which have dual functions.

- **Vibration sensor** - this sensor is used to detect whether an accident has taken place or not. It has a logic state indicator attached to it which is used to give the sensor a high/low signal showing accident/no accident. The sensor is connected to GPIO 13. The sensor is operated at 5 V.

- **Gps module** - this sensor is connected to RXD pin of the RPi board. It is used to get the GPS coordinates. This module provides us with the current location of the device in latitude and longitude.

- **Temperature sensor** - temperature sensor used in the project is connected to the 12 Bit A/D Converters with SPI(TM) Serial Interface which is further connected to CLK, MOSI, MISO, and

CS pin of RPi board. The temperature could be provided to the sensor by adjusting the low/high switch.

- **Compim** - In Proteus, COMPIM is used to model physical COM interfaces. It captures and buffers serial signals before presenting them to the electrical circuit. All serial data coming from the CPU or the UART model will be transmitted over the computer's serial ports. It is connected to the RXD pin RPi board.
- **16 x 2 LCD display** - for displaying any information on the screen. It is connected to GPIO 18, GPIO 22, GPIO 23, and GPIO 27. The data of 8 bits is sent to display twice since four pins are connected to the RPi board.
- **Relay module** - The module uses a DC motor and a Push-Pull four-channel driver with diodes(L293D). It is connected to GPIO 5 and GPIO 6.
- **Buzzer module** - this module is connected to GPIO 12.

4.5 Implementation

We have two sections in our project. First, we trained multiple ML models required in our project. We have implemented gaze detection. The dataset we used was [5]. The other module implemented for the project was eye blink detection for which we used a built-in dlib library function with pretrained face landmarks model to detect unique 68 facial landmarks, from which we extract the eyes landmarks and calculate the EAR to predict eyes closed or open and for what duration they are closed so that if they are closed for more than 3 seconds then the buzzer will go off and alert the driver of the drowsiness state. For mask detection module we have used **Opencv(cv2)** library in Python. We have also worked on detecting seatbelts by extending the Yolov3(you look only once) object detection model for our custom created object (seat belt) by training the model on custom created seat belt dataset of around 300 images of the seatbelt with labels as bounding boxes of the seat belt in each image. Detection of drowsiness of the driver using ML trained model has also been a major focus for the safety of the driver. The other section focuses on the simulation part where we are using a temperature sensor to detect the body temperature of the driver. A vibration sensor is used for the detection of a crash(accident). And in case a crash has been detected the location of the driver is read using the GPS module and is sent to the concerned family members. After the development of both the sections, we have worked on the integration of both the sections, i.e. ML trained model and Proteus simulation of the project. This is done using a compim which is used to model physical COM interfaces. It captures and buffers serial signals before presenting them to the electrical circuit. All serial data coming from the CPU or the UART model will be transmitted over the computer's serial ports. Figure 3

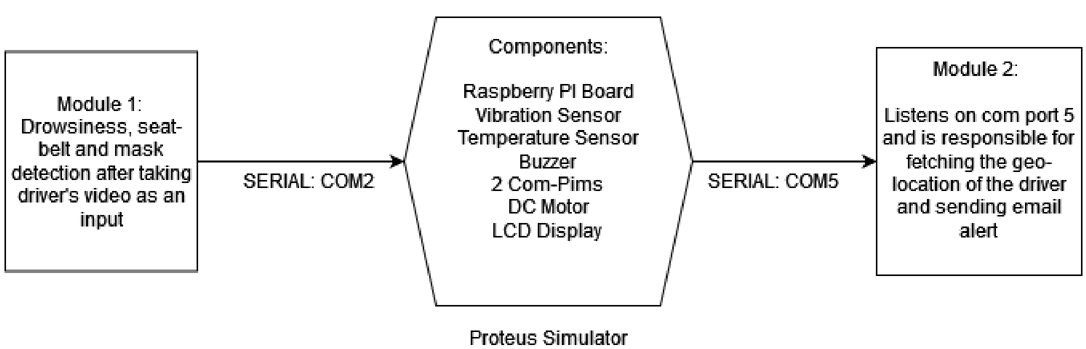

Figure 3: Diagram showing a high level understanding of the simulation

5 Results

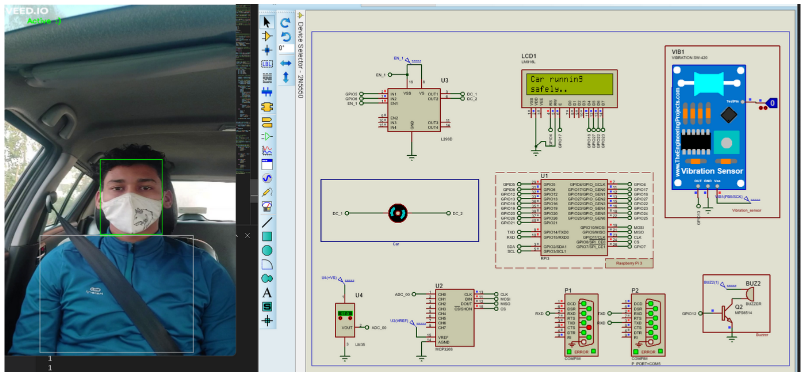

Figure 4: Car running safely,Mask and Seat Belt detected

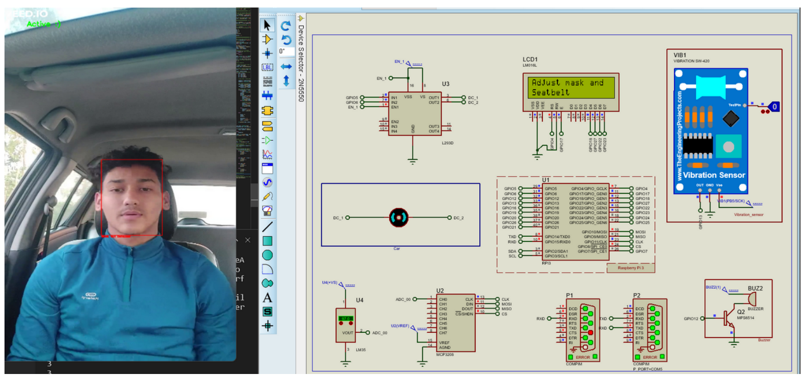

Figure 5: Output showing Adjust Mask and Seat Belt

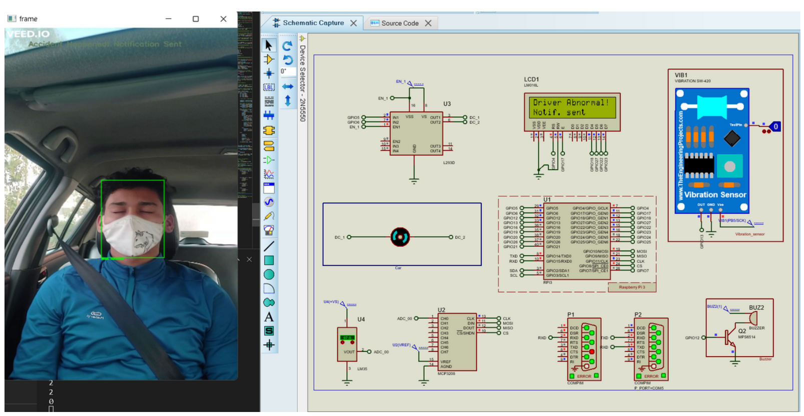

Figure 6: Output showing Abnormal driver and notification sent

6 Discussion and Future Work

We have successfully demonstrated the Safe-Vehicle system built on Proteus Simulator. The system uses various machine-learning models to detect face-mask, seatbelt, and gaze. The accuracy of these models can be improved by using a larger data-corpus and training for higher iterations.

The future work on this system can focus on improving the safety features of the car by using additional sensors that keep track of the health conditions of the driver. A body temperature sensor and heart sensor can be attached to the steering wheel which can provide additional information about the driving conditions.

Along with additional sensors, we can add a safety feature that brings the car to a halt and turns on the hazard lights when it detects that driver is drowsy or unconscious.

7 Conclusion

A safe car system has been successfully built as proposed by us. We are able to successfully detect mask, drowsiness and seatbelt worn by the driver. And in all the possible cases we are able to perform the required functionality. We are able to implement reverse communication from proteus to the python script and send email notification.

8 Individual Contributions

Team Member	**Contribution**
Gajender Sharma (25%)	1. Eye-blink Detection 2. Email notification system 3. Seat Belt detection
Kajal Sethi (25%)	1. Designing Schematic in Proteus 2. Location Detection
Pranshu Sahijwani (25%)	1. Drowsiness and Gaze Detection 2. Designing Schematic in Proteus
Shivam Tripathi (25%)	1. Face-mask Detection 2. Seat-belt Detection

References

[1] Common Causes of Road Accidents in India. https://www.dsclaw.co.za/articles/8-leading-causes-of-road-accidents-in-south-africa/.

[2] Eyenet Model for Gaze Detection. https://github.com/princeton-vl/pytorch_stacked_hourglass.

[3] Face Mask Dataset. https://github.com/prajnasb/observations/tree/master/experiements/data.

[4] Face Mask Dataset. https://www.kaggle.com/andrewmvd/face-mask-detection.

[5] Learning an Appearance-Based Gaze Estimator from One Million Synthesised Images. https://www.cl.cam.ac.uk/research/rainbow/projects/unityeyes/.

[6] Road accident statistics in South Africa. https://www.reliancegeneral.co.in/Insurance/Knowledge-Center/Blogs/Common-Causes-of-Road-Accidents-in-India.aspx.

[7] Smart Detection and Prevention system for Drowsy Driving. https://ilkogretim-online.org/index.php/fulltext/218-1618312234.pdf?mno=81254.

[8] Synthetic Gaze and Face Segmentation. https://www.kaggle.com/allexmendes/synthetic-gaze-and-face-segmentation.

[9] VEHICLE SPEED CONTROL SYSTEM USING GSM/GPRS. http://www.ijstr.org/final-print/jan2014/Vehicle-Speed-Control-System-Using-Gsmgprs.pdf.

[10] YOLOv3: Real-Time Object Detection Algorithm. https://viso.ai/deep-learning/yolov3-overview/.

[11] Auni Syahirah Abu Bakar, Goh Khai Shan, Gan Lai Ta, and Rohana Abdul Karim. Iot—eye drowsiness detection system by using intel edison with gps navigation. In Zainah Md Zain, Hamzah Ahmad, Dwi Pebrianti, Mahfuzah Mustafa, Nor Rul Hasma Abdullah, Rosdiyana Samad, and Maziyah Mat Noh, editors, *Proceedings of the 10th National Technical Seminar on Underwater System Technology 2018*, pages 485–493, Singapore, 2019. Springer Singapore.

[12] Anil Kumar Biswal, Debabrata Singh, Binod Kumar Pattanayak, Debabrata Samanta, and Ming-Hour Yang. Iot-based smart alert system for drowsy driver detection. *Wireless Communications and Mobile Computing*, 2021, 2021.

[13] Arjya Das, Mohammad Wasif Ansari, and Rohini Basak. Covid-19 face mask detection using tensorflow, keras and opencv. In *2020 IEEE 17th India Council International Conference (INDICON)*, pages 1–5, 2020.

[14] Chengcheng Fan, Xiang Liu, Ronghua Ling, and Boyu Si. Application of proteus in experimental teaching and research of medical electronic circuit. In *Proceedings of the 2018 3rd International Conference on Modern Management, Education Technology, and Social Science (MMETSS 2018)*, pages 512–515. Atlantis Press, 2018/11.

[15] Sahil Haria, Shubham Anchaliya, Vaibhav Gala, and Tina Maru. Car crash prevention and detection system using sensors and smart poles. In *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, pages 800–804, 2018.

[16] Seok-Woo Jang and Byeongtae Ahn. Implementation of detection system for drowsy driving prevention using image recognition and iot. *Sustainability*, 12(7), 2020.

[17] Alexey Kashevnik, Ammar Ali, Igor Lashkov, and Nikolay Shilov. Seat belt fastness detection based on image analysis from vehicle in-abin camera. In *2020 26th Conference of Open Innovations Association (FRUCT)*, pages 143–150, 2020.

[18] Qi-Chao Mao, Hong-Mei Sun, Yan-Bo Liu, and Rui-Sheng Jia. Mini-yolov3: Real-time object detector for embedded applications. *IEEE Access*, 7:133529–133538, 2019.

[19] Patchava Vamsikrishna, Sonti Dinesh Kumar, Shaik Riyaz Hussain, and K. Rama Naidu. Raspberry pi controlled sms-update-notification (sun) system. In *2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, pages 1–4, 2015.