



**VIT<sup>®</sup>**

**Vellore Institute of Technology**

(Deemed to be University under section 3 of UGC Act, 1956)

Winter Semester 2020-2021  
ECE3502 - IoT Domain Analyst  
Project Report

# Drowsiness Detection and Rest Stop Suggestion

18BEC0002 - SABARI MATHAVAN R B  
18BEC0711 - NIKHIL SUBRAMANIAN  
18BIS0019 - ROHITH KANNA D

## **Introduction:**

With the growing economy in this fast paced world, the workload on drivers who do long delivery have been increasing tremendously. This is singly due to very short deadlines for the delivery completion. The drivers to earn money are made to drive long distances in a short period of time which may become disastrous. These are mostly caused by human errors like rash driving, DUI and tired and sleepy drivers. This will cause accidents on roads causing a problem to fellow travellers and the people who require the goods being transported.

## **Abstract:**

To fix this, we have concentrated on addressing the problem of tired and sleepy drivers. Due to these very close delivery times, drivers tend to not sleep and drive. Our project will notify the driver and the manager that the driver has reached his limit and is required to take a rest stop near him. To save the hassle of finding a stop nearby, our project will also give a list of nearby rest stops for the driver to choose from. On choosing, it will set the GPS route to the rest stop. While doing this, the manager of the logistics company will be emailed the location with the timestamp of the driver who was detected to be tired. The manager can then track the driver by using the Driver's Lounge app. This app will take the coordinates of the driver from the Firebase cloud which stores the information of the driver, vehicle number, location, timestamp when the driver is detected tired.

## **Literature survey:**

### **A brief introduction to OpenCV**

This paper aims to introduce and make a reader immediately familiar with the foundations of OpenCV, without having to read extensive reference manuals and books. OpenCV was first created by Intel throughout the past decade and is an open source library for Image and Video analysis. Several programmers have been helping with current advances in libraries since then. In 2009 the last major upgrade (OpenCV 2) included extensive C++ interface modifications. The library now has >2500 algorithms optimised.

It is widely utilised worldwide, with >2.5M and >40K downloads in the user group. Whether a newbie programmer for C++ or a professional development software developer, without knowledge of OpenCV, the primary library material should be of interest to graduates and researchers in the fields of image processors and computer vision. Many books on the topic of OpenCV are needed to grasp every aspect in the library. However, after understanding some principles of OpenCV in this article, it should be easy to study such more thorough content.

### **Design of DroDeASys (Drowsy Detection and Alarming System)**

The paper outlines a non-intrusive method to the Drowsy Detection & Alarm System that was created. The device is primarily designed to identify drivers napping at night. The technology employs an infrared camera that directly points to the driver's face, monitoring driver's eyes to identify weariness. In this situation, a warning signal to inform the driver is given when weariness is identified. This study explains the methods used to identify somnolence. Depending on whether the eyes are open to a particular number of frames, the choice is whether or not the driver is asleep.

### **Selecting Location of Retail Stores Using Artificial Neural Networks and Google Places API**

The choice of retail shops sites is a sort of decision issue, where the researcher uses statistical or judgement projections for greatest profit or least costs. In many circumstances, survey data, or data obtained for another business form the basis of statistical estimates that are unreliable since the survey may not measure prospective demand or the alternative shop used for comparison may not be sufficiently similar. In addition, it might be costly to estimate the possible demand for a non-existent store. Google Places API is a software library offering a nearby

search tool that reports various types of organisations, organisations, etc.

In this research they propose a strategy based on an assessment of the relationship between the environmental qualities and financial rating of current retail establishments.

They point out that neural networks can disclose such a relationship that they can use integer, true, and binary models, as well as non-linearity. They apply their solution to the retail chain's location selection problem in Turkey (Note 1), where there are 144 customers. Expert comments in the organisation show us that in comparable scenarios the applicability of this strategy is great.

## **Application of Firebase in Android App Development-A Study**

A great deal of data, including movies, photos, music, texts, files and other arbitrary forms have grown more dependent on the web app. The management of unstructured databases is problematic for the Relational Database Management System. A relatively recent technique to manage enormous quantities of unstructured data is Firebase. Compared to RDBMS it is really quick. The purpose of this article is to apply Firebase to Android and to familiarise it with its principles, terminology, benefits and constraints. The article also tries by constructing an Android app to show some of the functionalities of Firebase.

## **Real-time Communication Application Based on Android Using Google Firebase**

Connection is incredibly critical in the world today and it is crucial to maintain this communication in real time as our lives progress more quickly. In order to keep this in mind, files and messages should be able to be sent quickly or with little delay, depending on the broadcast media. To be functional for such a system, a database must be installed that updates all data transmitted in real time. Google Firebase offers such a real-time database server and a plethora of other features and Firebase allows us to very easily construct communications based apps.

This article proposes a system that would enable two network users to communicate in real time text messages and data such as photographs, music, videos, texts over the internet. You utilise the Android and Google Firebase operating systems to manage the backend of communication, emphasising the different aspects of both the operating system and the service.

## Experience of Developing a Mobile Application Using Flutter

The creation of mobile applications means that two versions are always developed—for Android and iOS. Flutter, Google's latest technology, enables you to create one code apps for both systems. It also provides the opportunity for native applications to be created. Their experience utilising this technology to construct a mobile application for investment hotels is discussed in this article.

### Proposed system:

The system is a handy device which can be placed on the dashboard of the truck with the camera facing the driver. This device will consist of a Raspberry Pi, with a camera and a buzzer. The device will also require constant internet connection so a SIM module will also be used. Rest of the requirements are python code and cloud storage. To improve the stability of the camera a gimbal may be used too. This Raspberry Pi will send data to firebase cloud storage and use google API for finding rest-stops

### Methodologies:

#### Drowsy Detection using OpenCV:

The code uses the dlib library to give the code ability to put up landmarks on the image and track it. Using the dat file, it can detect and track a face and its features. There are a total of 68 landmarks out of which 12 landmarks denote eyes. (6 for each eye) Using these points, the Euclidean distance can be calculated and a ratio can be calculated which shows the size of the shape obtained by joining the eye landmarks.

So first using OpenCV, the camera is operated. This colour image is converted into **grayscale** for better detection of the landmarks. For each frame the eye area is calculated. When the area is less than a specific value, it will mean that the eye is closing so it detects **"DROWSY"**.

## Mapping the locations:

The locational coordinates of a place is always represented as a set of 2 values - latitude, longitude. These 2 dimensional coordinate values can represent any place on earth. The longitudes range from -180 to +180 from prime meridian. The latitudes range from -90 to +90, with 0 as the equator. These values are represented in the form of degrees, minutes and seconds. But for our use we are converting the data into decimal format. To determine the locational coordinates from the system we use Windows runtime API. It provides the results within a distance error of 10m if GPS is being used, or upto 500m if Wi-Fi is being used for tracking. The Windows API along with its Geolocator function provides the results in the form of geographic locational coordinates (decimal).

To collect information about the nearby resting spots, or restaurants for resting/refreshments, Places API of Google Developers. The Places API is a service that returns information about places using HTTP requests. The locational coordinates obtained from the previously mentioned steps are passed as parameters to the API call, along with the place type attribute. The search results are returned in the form of JSON text. The obtained text sets are further processed to get the required parameters - location name, coordinates.

The obtained locations along with the distance of the place from the current location of the user is displayed and the user is made to make a decision on which place he wants to visit. Once the user selects the required location, a map is configured with the source (user's location) and destination with marker, using gmaps. "gmaps" is a Jupyter plugin for embedding Google maps in Jupyter notebooks. It is designed to help visualize and interact with geographical data. Routing between the source and the destination is done for a better visualization.

The final map, after adding of direction layers, is displayed to the user to navigate. Along with the map the distance of the place from the current location of the user, the approximate time taken to reach the destination and the address of the destination is also provided.

## Drivers Lounge:

With both the backends finished, we just need a user interface to show the drivers the various places that are recommended to be stopped in. That's where our app comes in. Drivers Lounge is to show the details of the drivers. This is done by getting the data from Cloud Firestore that was stored in the previous model, and using that to display driver details such as the driver's name, plate number, and when the data was received. The location is also displayed on the map.

This is done using Flutter as it provides a limitless UI with ease, as well as being cross-platform. We then had to configure Maps SDK for Android and for iOS through the online Google Cloud Console and add that to our app as well.

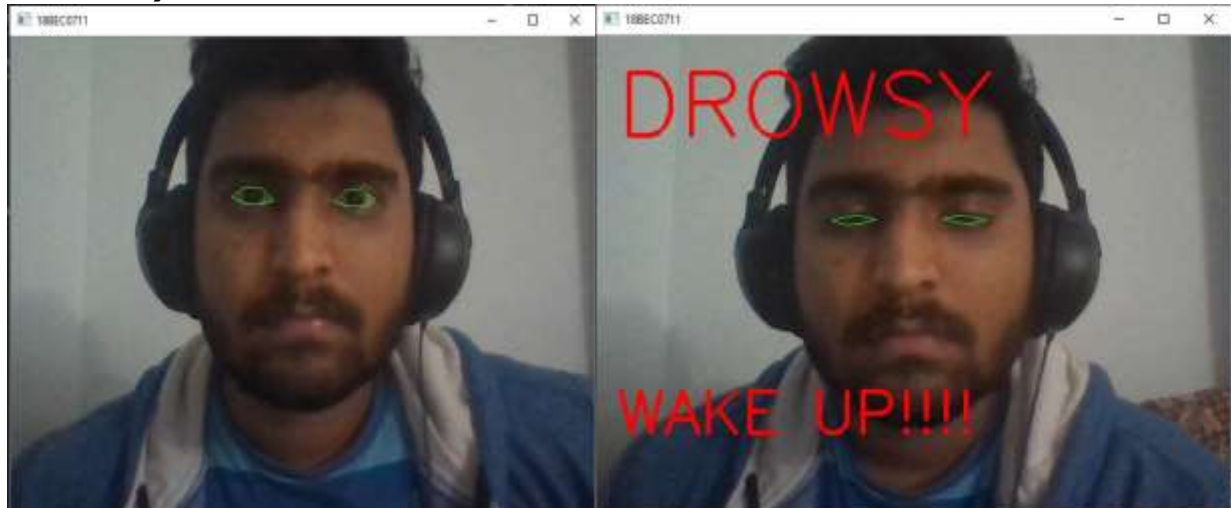
## Cloud Firestore and Email:

Storing data is important for future uses. For example, to have a track of tiredness record of each driver and the time record of tiredness for each driver. So, to analyse the records of each driver we store values such as the driver's name, vehicular registration number, timestamp and locational coordinates of the place where tiredness was detected. To store these data values, Google Cloud Firestore was used. Cloud **Firestore** is a NoSQL document database that lets you easily store, sync, and query data for your mobile and web apps - at global scale. Google Cloud Firestore is configured with the python file using serviceAccountKey, which is a JSON file provided by Firestore to access a particular database account. Data is sent over to the firestore, and recorded under the database table named 'records'.

For further notifications, we have enabled email sending option to the manager. The email is sent using SMTP protocol. The Simple Mail Transfer Protocol (SMTP) is an internet standard communication protocol for electronic mail transmission. Python uses the "smtplib" library to send emails from a specified user to another user. Values like the vehicular registration number, the driver's name, the locational coordinates and time of fatigue detection are sent in the mail as the message.

## Results:

### Drowsy detection:



### Rest stop suggestion upon drowsy detection:

```
PS C:\Users\sabar\Desktop\inf drowsiness detection> & "C:\Users\sabar\AppData\Local\Programs\Python\Python39\python.exe" "C:\Users\sabar\vscode\extension\inf-python-python-2021.5.8\2521120\pythonfiles\lib\python\idebuggy\launcher" "9903" "-c" "C:\Users\sabar\Desktop\inf drowsiness detection\drowsiness_detection.py"
-> Starting Video Stream
call
latitude: 9.916665
logitader: 78.0099125
1 --> The Gateway Hotel Panmalai, Madurai
2 --> Astoria Hotels
3 --> Hotel Sree Devi
4 --> Hotel Germania
5 --> Hotel The Rock
6 --> Regency Madurai by GRT Hotels
7 --> OYO 7355 Govinilla
8 --> Madurai Ladies Hostel
9 --> Jene's Ladies Hostel
10 --> Pandian hotel
11 --> MESDOM ASSET PROMOTERS INDIA LIMITED
12 --> Coral Shelters Andalpuram
13 --> PH Residency
14 --> Veera Anshaa Service Apartments
15 --> Hotel Sri Arulmuthu Residency
16 --> Ra Global Residences Rooms (odge Main City Madurai
17 --> Hotel Archana Veg & Rooms
18 --> OYO 5044 The Rock
19 --> Nest Serviced Apartments
20 --> LIVING ORACLE

Choose a location
13

Figure(layout=figurelayout(height='400px'))

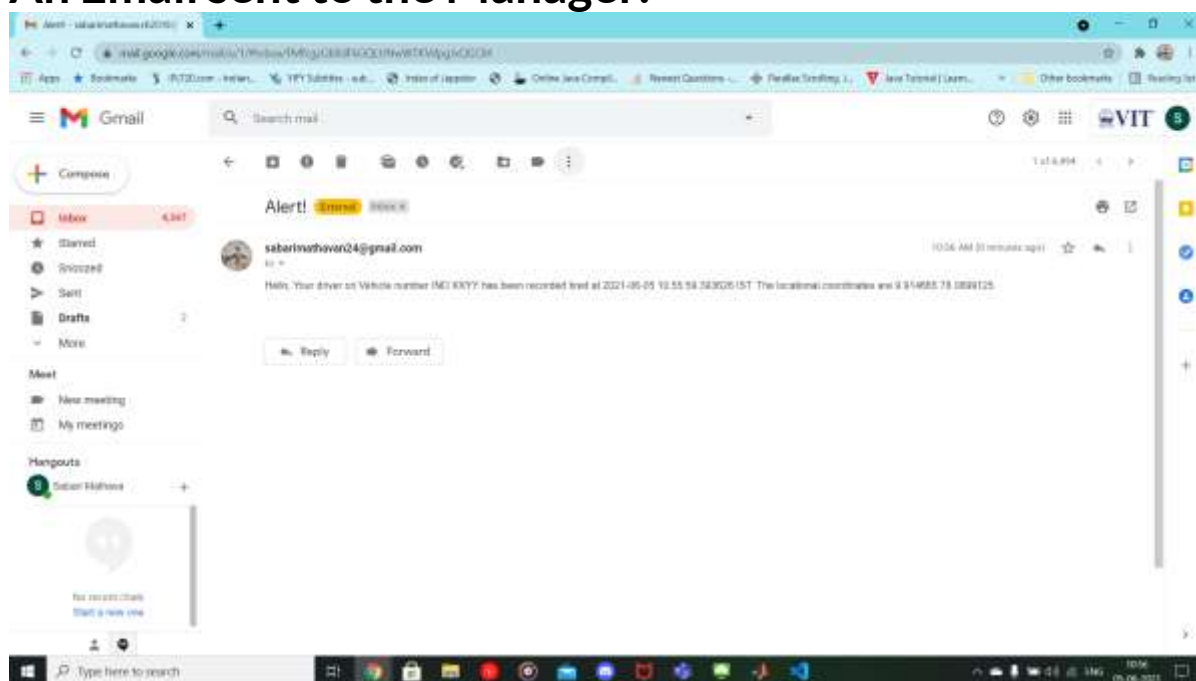
PH Residency
3.2 km
11 mins
1, Pillayar Kovil St, Mahakood Palayam, Madurai, Tamil Nadu 625016, India
PS C:\Users\sabar\Desktop\inf drowsiness detection>
```



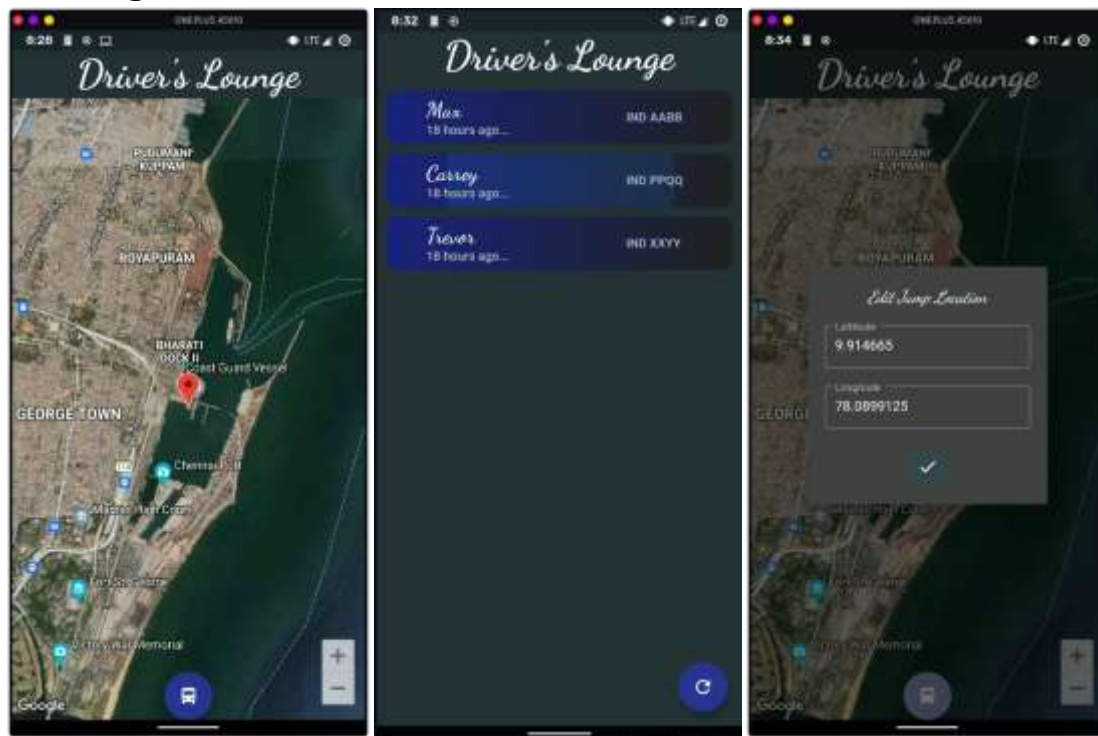
## Map direction to the chosen rest stop:



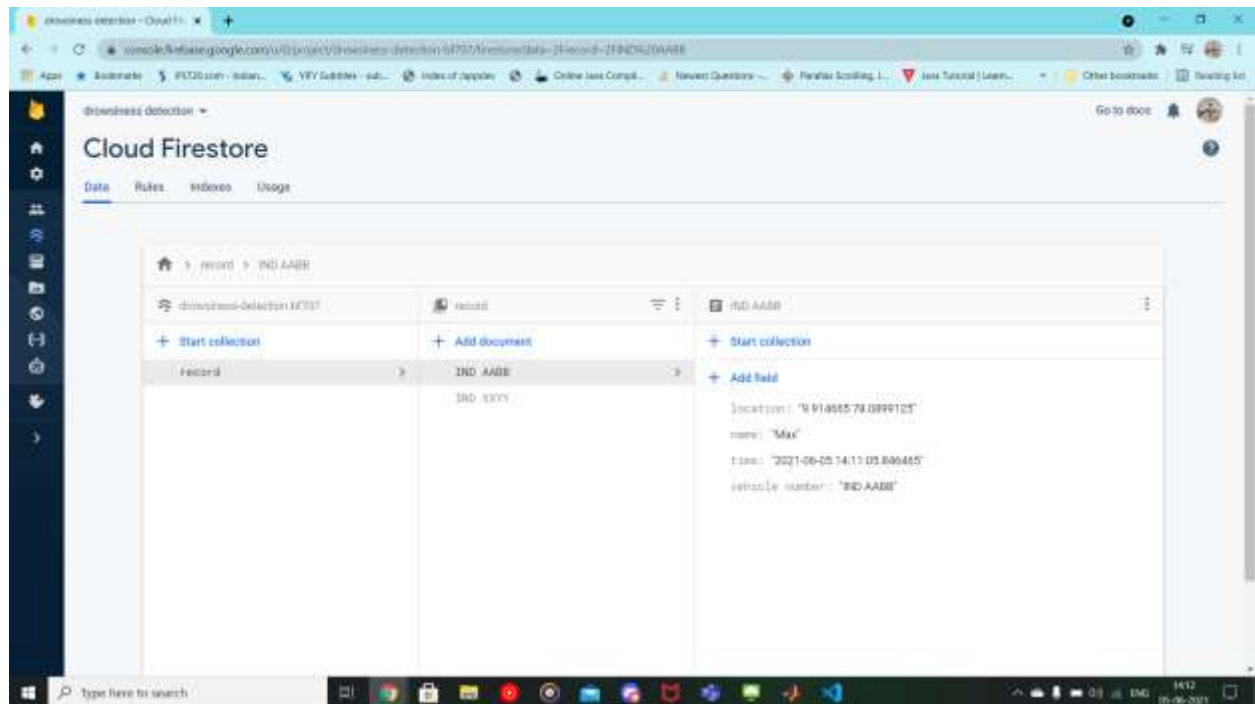
## An Email sent to the Manager:



## Manager's app to track all driver's location:



## Driver location fed into Cloud:



## Conclusion and future works:

A system which can detect the driver's drowsiness/tiredness and suggest res-stop while intimating the situation to the manager has been created. This device can finally be used to reduce road accidents caused by fatigue drivers.

There are many future scopes to this project:

- This project can be extended into cars with a driver's lounge accessible to family members.
- With the addition of algorithms like image registration, the system can become better because it can record the face better.
- Implementing an alarm system when the device loses connection will make the device better because when driving long distances the network connectivity will be a problem.
- Presently our project uses only eyes as a parameter for drowsy, adding another parameter will solidify our project's work as a drowsy detector.
- With the era of automatic driving approaching, this system can be installed to stop at a rest stop when the driver falls asleep by feeding the rest stop location directly into the car's navigation system.
- By adding a crash detector, this device can be modified into a first responders signal beacon in the event of an accident.

And more additions can be done.

## References:

[1] A brief introduction to OpenCV

I. Culjak, D. Abram, T. Pribanic, H. Dzapo and M. Cifrek, "A brief introduction to OpenCV," 2012 Proceedings of the 35th International Convention MIPRO, 2012, pp. 1725-1730.

[2] Design of DroDeASys (Drowsy Detection and Alarming System)

Juvale H.B. et al. (2009) Design of DroDeASys (Drowsy Detection and Alarming System). In: Ao SI., Rieger B., Chen SS. (eds) Advances in Computational Algorithms and Data Analysis. Lecture Notes in Electrical Engineering, vol 14. Springer, Dordrecht. [https://doi.org/10.1007/978-1-4020-8919-0\\_6](https://doi.org/10.1007/978-1-4020-8919-0_6)

[3] Selecting Location of Retail Stores Using Artificial Neural Networks and Google Places API

Mehmet Hakan Satman & Mustafa Altunbey. Online Published: January 21, 2014  
doi:10.5539/ijsp.v3n1p67 URL: <http://dx.doi.org/10.5539/ijsp.v3n1p67>

[4] Application of Firebase in Android App Development-A Study  
Chunnu Khawas & Pritam Shah. International Journal of Computer Applications  
(0975 – 8887) Volume 179 – No.46, June 2018

[5] Real-time Communication Application Based on Android Using Google Firebase  
Nilanjan Chatterjee, Souvik Chakraborty, Aakash Decosta & Dr. Asoke Nath

[6] Experience of Developing a Mobile Application Using Flutter  
Kuzmin N., Ignatiev K., Grafov D. (2020) Experience of Developing a Mobile  
Application Using Flutter. In: Kim K., Kim HY. (eds) Information Science and  
Applications. Lecture Notes in Electrical Engineering, vol 621. Springer, Singapore.  
[https://doi.org/10.1007/978-981-15-1465-4\\_56](https://doi.org/10.1007/978-981-15-1465-4_56)