Driver Evaluation App

CSE499A

By

Md Shafaqul Islam Hamim

ID: 1811762642

Muid Ahmed Alvi

ID: 1812880642

Shahida Akter Chowdhury

ID: 1811205642

Electrical and Computer Engineering Department

North South University

Dhaka, Bangladesh

September, 2021

Contents

Introduction:

Problem Area:

In our country the number of people use car of 4,471,625 registered vehicles in our country, of which only 370,519 are registered as ‘Private Passenger Car’ followed by 105,896 ‘Microbus’ and 66,219 ‘jeeps’ (Data from Bangladesh Road Transport Authority (BRTA) shows this) And (According to BRTA statistics) there were 2.92 lakh drivers with heavy and medium driving license in the country who drove vehicles such as buses, minibuses, trucks, pickup, and covered, cargo and delivery vans.

So, it’s quite common that most of the people use cars and have drivers but every day they face a lot of problems and want to know how their driver is treating the car. Moreover, they want to know if their car in on the right hands or not. The owners of the car want know the state of the car and how it’s been driven even if they are not in it.

Importance of the Problem:

To ensure the safety of the car and the passenger in the car a good driver is the most important necessity. Because in our country car accidents happens every day as many as 6,686 people lost their lives and 8.600 were injured in a total of 4,891 road accidents in 2020 in Bangladesh. Many of these accidents happen when the driver is not good enough to drive the car. Also, there are many accidents that happen in the highways when the driver fell asleep while driving. Also, an owner of a car wants to know about his car how it is driven. When the owner or the stakeholder always knows what is happening in the car they can monitor over the driver as well as know how disciplined the driver is, the driver will be careful about his driving. That should certainly decrease the rate of accidents and car damage.

Formal Explanation of the Problem:

The problem we are focusing on is related to Vehicle safety which is deeply related to the driver’s performance. For this project, we are going to focus on evaluating a driver’s performance and provide the owner information through mobile app. The evaluation criteria will be, Dizziness detection, Safety belt, Side mirrors, overtaking, Speeding, Frequent breaking, pedestrian interaction, Parking, Cleaning the car, Routs knowledge, Oil report (Described in the System Design Section).

Scope of the Problem:

Our Project goal is to create a system that will provide a car owner the information about his car and the performance report of its driver.

End of the project we will have a system installed in the car which will provide information to the server and then it will be collected by the mobile app and then by processing the information gotten the app will show the data to the owner of the car.

Tasks for this problem will be

I) Deriving the data from the car
II) Sending it to server
III) Building the mobile application
IV) Fetching data from server
V) Processing Data
VI) Showing it to the user.

The cost is estimated to be 20,000 BDT approximately for buying the required tools.

Our estimation about the completion of the project is set to be December, 2021.

Strategies That Can be Used:

Here our main problem is to make personal vehicle safer. In Bangladesh maximum people appoint driver for their cars. But the problem here is how they are safe behind the driver. there are some strategies to evaluate a driver.

1. A driver can be tested by a someone who is expert at this. They can have a session behind the driver and give him rating.
2. The owner of the car can observe the driver for few days and fillip a form in a mobile app. The app will analysis the data and will show drivers score.
3. We can observe the driver's car interaction from OBD 2 port. We will get data of vehicle usage and we can analysis the data, we will also have a dash camera to detect driver's behaviors during driving. we will have an app which will analysis data and give result to owner through the app UI.

Our Strategy:

Here we found no. 3 strategy more accurate to solve the problem. Because in this strategy driver would be monitored simultaneously. For this driver would try to be as aware as possible which will get more safety for the owner. And we can set some alert system to notify the owner immediately for greater issues.

Literature Review

Literature we have reviewed are:

1. Various sensors have been studied to perform lane-position determination. Examples of these include: * camera and vision sensors; * internal vehicle-state sensors; * line sensors; * LASER radio detection and ranging (RADAR) sensors; * global positioning system (GPS) sensors. While LASER RADAR, line, and GPS sensors can perform extremely well in certain situations, vision sensors can be utilized to perform well in a wide variety of situations. LASER RADAR sensors are useful in rural areas for helping to resolve road boundaries, but fail on multilane roads without the aid of vision data. Line sensors, while accurate for current lateral position, have no look ahead and cannot be used well for trajectory forecasting, which is needed to compute metrics such as time to lane crossing (TLC). GPS, especially differential GPS (dGPS) , can provide accurate position resolution, but this

requires infrastructure improvements to achieve these accuracies, and to rely on map data that may be outdated and inaccurate. Vision sensors can provide accurate position information without the need for external infrastructure or relying on previously collected map data. In the situations where vision sensors do not perform well (i.e., extreme weather conditions or off-road conditions), the vision data can be fused with other sensor modalities to provide better estimates. This makes vision sensors a good base on which to build a robust lane position sensing system. Because of these reasons, this paper will focus mainly on vision sensors augmented by vehicle-state information obtained from the in-vehicle sensors.

2. VEHICULAR DATA COLLECTION DESIGN Our vehicular data collection consists of a backend that directly communicates with in-vehicle sensors and collects sensor data. This backend is part of a distributed testbed that allows access to smartphone sensor data in a secure and performance isolated way. Furthermore, data can be backhauled to a centralized server for permanent data storage. This server also serves as a frontend to make the data available to experimenters interested in traffic monitoring, etc. In this section we describe the design of our platform. Distributed Smartphone Sensor Testbed: We deploy our platform on a distributed testbed that provides an efficient programming environment for our data collection. Experimenters can implement automated experiments and collect data from accelerometer, GPS, WiFi, camera, and other sensors on mobile devices that are owned by the general public. Conducting these experiments requires minimal power and networking resources, used in a non-intrusive manner. Our testbed employs a light-weight sandbox to limit the amount of storage, network, memory, battery, and CPU resources used on a mobile device. To protect end users' devices from malicious attackers, our sandboxed programs are securely isolated from other programs on the same device. Data Collection Overview: The process of deploying an experiment on the testbed and then collecting data from vehicular sensors on end user devices is as follows. Device owners install a testbed app on their devices. Since we are interested in vehicular data, the target group of device owners are also vehicle owners. These owners simply insert their WIFI On-Board Diagnostics (OBD) sensor into.
3. Hardware-Based Phase This section provides a concise overview of the test experiment design for designing a hardware alarm system based on data analysis of driving behavior, including various experiments and scenarios. A selection of standards and metrics were taken from prior methodologies by analyzing papers and information about the collision warning system. Few studies have been directed in hardware development for warning systems based on driving behavior context. From this point, the hardware side serves two main functions. The first is the process of collecting and exchanging data between vehicles, and this is presented in the following subsection. The second aspect is related to the process of data collection of driving behavior and how to use the available devices and

software. Table 1 indicates several factors that helped to choose the optimum device that offers the required objectives. Table 1 clearly shows a few studies have been directed towards either data exchange in V2V or related to driving behavior except for one article that combined these two terms. They state that the driving behavior-based alert system's performance was only tested in the lab with actual vehicle data. As a result, more analysis is requested to assess the output using various driving data and alerts that arise when driving in a real vehicle. In addition to that, reference indicated the possibility of using wireless communication module nRF24L01 to send and receive data for warning based on Bluetooth and ultrasonic sensors. Finally, reference proposes a driving behavior analysis approach based on OBD-II information to collect vehicle activity information via OBD-II adapters. Vehicle activity information may include vehicle speed, throttle location, engine RPM, and measured engine load, then complete data preprocessing and driving behavior modeling in offline mode using MATLAB. Hence, using wireless technologies for information exchange has the potential to affect driver behavior in the direction of better driving safety and fewer road injuries. During the implementation process of this scheme, we concentrate on a new aspect called driving behavior to overcome the problems and limitations of previous researchers.

4. For our prototype platform, the OBD-II ELM327 WiFi sensor was used. In order to facilitate communications between sandboxed programs on a smartphone and the OBD sensor, we developed the obdlib library, which implements the standard OBD communication protocol. We used a TCP connection between the smartphone and the OBD sensor via the default WiFi gateway IP address assigned to the OBD sensor3 . In addition to the TCP connection wrapper, obdlib includes OBD standard parameter ID (PID) commands that For these reasons we decided to use a non-relational database, MongoDB, for storing sensor data. MongoDB has a Binary JSON (BSON) document-style structure identical to that, albeit converted into binary as its name suggests. This provides the scalability that is crucial to our system and that allows dynamic storage of new sensors, as needed. Non-primitive data types can also be stored more easily because no column structure exists in MongoDB that enforces model schema or data typing. Additionally, because our sensor data is already in JSON format, MongoDB is the most convenient storage option. For the purposes of securing stored sensor data and easing the flow of an experiment, this implementation is made transparent to the user by our website's interface. Within the database, individual devices are distinguishable to an experimenter by using a stored, unique identifier. We chose to use the device's IMEI number because it is a 15-digit ID that is unique to each device's hardware. Although the SIM number might be a more familiar choice, we decided not to use it because the devices used in our testbed are not required to have data plans. Experimenters can instead opt to run their experiments solely through WIFI networks.

5. Establishing Communication Between the OBD II Reader and the ECU Firstly communication between the OBD II reader subsystem and the vehicle ECU needed to be verified. A Freemartins OBD II emulator, was used for testing purposes. The emulator implements three OBD II protocols (CAN, ISO 9141-2 and ISO 14230-4. It has an OBD II 16 pin connector similar to an actual OBD II compliant device and was powered from an AC to DC power supply connected to the mains. A USB to serial convertor was used for communication between the PC and the ELM327 IC on the reader. The speed and MAF parameter values were initially configured on the emulator. AT and OBD II commands were sent from the MCU on the reader to the emulator to initialize communication. The data received from the emulator was then displayed on the terminal program, Termite (which was set at a baud rate of 38400 Kbps).

Source links:

1. https://sci-hub.se/10.1109/TITS.2006.869595

2. https://sci-hub.se/10.1109/SAS.2015.7133607

4. https://sci-hub.se/10.1109/SAS.2015.7133607

5. https://sci-hub.se/10.1109/JSEN.2016.2631542

System Design:

Systems design is the process of defining the architecture, product design, modules, interfaces, and data for a system to satisfy specified requirements. Below we have provided a diagram of our system design.

As we discussed earlier, our project is about vehicle safety. For this we are planning to evaluate a driver's performance. To do such we have decided to take inputs from the user as well as the vehicle itself.

Components of our system are a car, OBD ii port, a Raspberry pi 4, Dashcam, Server, a mobile app and, the user.

The input system of this application will be taken in two sections.

I) System input
II) End user input.

System Input: In our system, system input means taking input from the car and camera. We will take speed, acceleration, seatbelt information, oil level information, RPM, and Engine Information from the car. From the camera which will be facing the driver from the front will provide video input to the Raspberry.

End User Input: The app UI will provide necessary forms to the user and the user will give input accordingly to evaluate the performance of the driver.

For the evaluation we will provide 100 points to the driver every month. And for each time any problem is detected his points will be deducted and the remaining points in the end of the month will be his final points for the month. And we will provide a list of months with the points and driving history.

Evaluation Criteria:

1. Dizziness -

Dizziness while driving is common in Bangladeshi drivers. As you might expect, having such an experience while in control of a vehicle can be extremely dangerous, both for himself, any passengers in your vehicle, drivers of other vehicles and pedestrians. The end user will automatically get a notification while frequent dizziness and yawn is detected. Each time it is reported 10 points will be deducted.

Derivation: From dashcam using computer vision algorithm and OpenCV, dlib.

2. Safety belt -

A seat belt, also known as a safety belt, or spelled seatbelt. A vehicle safety device designed to secure the driver or a passenger of a vehicle against harmful movement that may result during a collision or a sudden stop. When in motion, the driver and passengers are traveling at the same speed as the car. Detection of this will be in two steps, from the car and also from the camera using detection algorithm and also the speed of the car will be taken into consideration. If the speed>15km/h then it will report. If in both seatbelt is not detected then the information will be sent to the server and then pushed into the mobile app. This will deduct 5 points for each time.

Derivation: From the system of the Car, and detection algorithm, OpenCV, dlib.

3. Side Mirrors -

Mirrors allow you to observe what is happening around your car. They are your most important visual driving aid and are vital for safe driving. Their purpose is to let you know what is happening behind them, which is just as important as knowing what is happening in front. Detection of the side mirror checking will be done in two steps from the camera we will detect where the driver is looking and then crosschecking it from the car’s corresponding indicator is on or not. Will be deducted 2 points each time.

Derivation: Using Computer Vision, dlib and car.

4. Overtaking-Vehicle –

Overtaking is when you approach from behind and pass a vehicle traveling in the same direction. Most drivers and riders consider overtaking to be crossing to the 'wrong side of the road to pass a vehicle in front. 5 points will be deducted after the end of the month based on the can owner's input.

Derivation: User Input.

5. Speeding uncomfortable -

The sickness affects people on long road journeys. It's caused by **repetitive movements**, such as accelerating and decelerating and turning around a corner at speed, which can disturb the inner ear. In addition to helping, you hear the sound, the inner ear helps you balance. Acceleration/deceleration<0.9 m/s^2 would be comfortable. So, if this kind of acceleration is crossed 3 times in 10 minutes, we will send information to the server. 3 points will be deducted for each time.

Derivation: From the Car's System

6. Pedestrian interaction-

when driving he/she must have overlooked on the road that any kind of person or animal is passing the road or not. Deduction of 5 points based on user input each month.

Derivation: User Input

7. Parking-

parking a car in a proper way is very important in any place whether it's in village or town. Based on user input 3 points deduction each month.

Derivation: User Input

8. Clean-

Having a clean car is a key part of the Lyft experience. It's relaxing for passengers (and you!) to ride in a spotless and nice-smelling car. You'll be rated accordingly. You can check out our 5-star driving tips in the Learning Center tab of your Lyft Driver app for cleaning tips from some of our top drivers. Deduction of 5 points if user is not satisfied each month.

Derivation: User Input

9. Following the maintenance of vehicle –

Derivation: Reports on any malfunction in the car from the car than asking the owner whether it was reported by the driver.

10. Routs knowledge –

It means having adequate information and knowledge about the roads and routes of the area. Deducts 5 points.

Derivation: User Input.

11. Skilled-

One of the most important qualities of a good driver is that he or she should be skilled enough to handle any situation on the road. A good driver should have the skills to control the vehicle in every situation.

Derivation: User Input

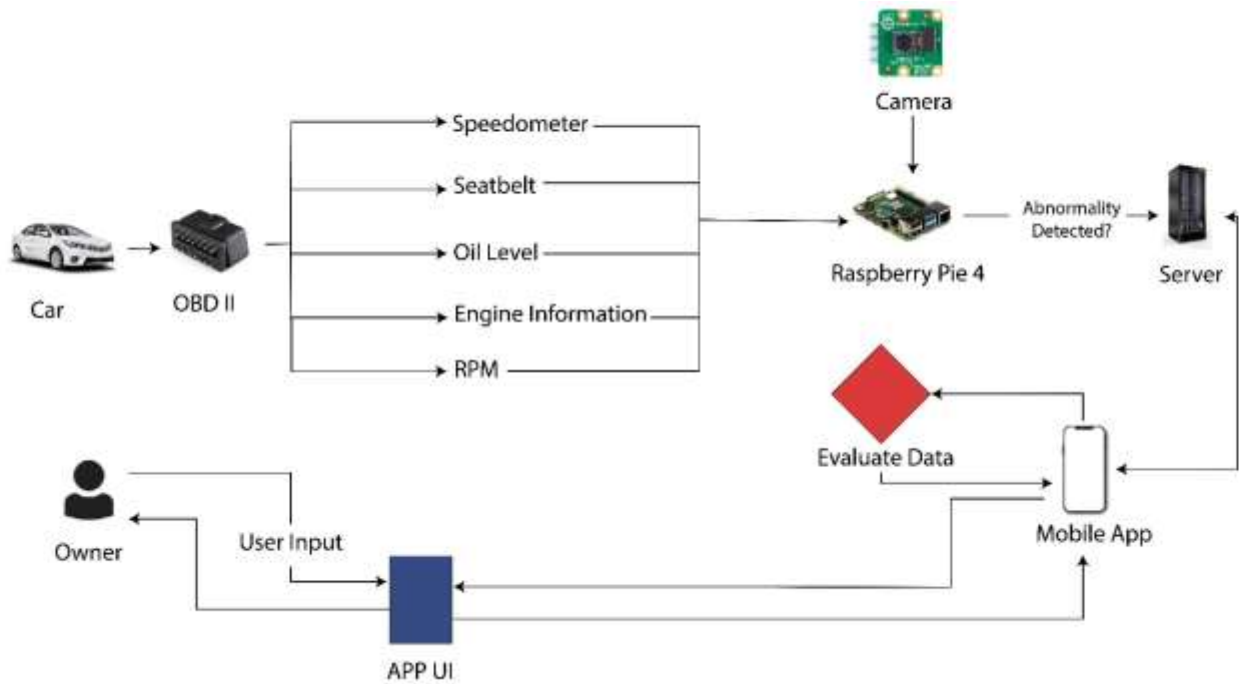
12. Oil report –

This is only for the owner to know how much oil is in the car. This will be updated real time whenever the user wants to know the level of oil in the car.

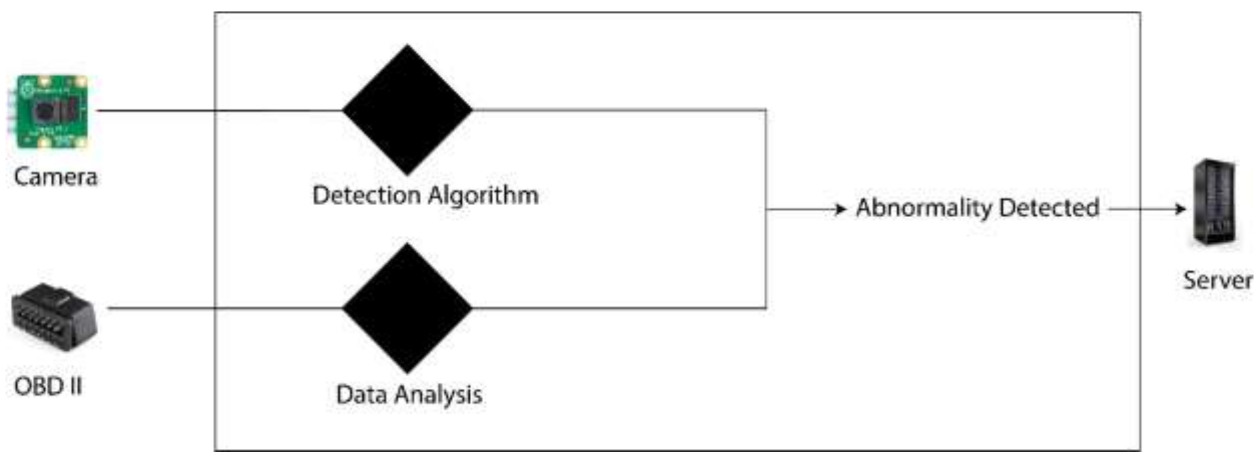
Derivation: From the System

The Flow of Data:

We will derive data from the car using obd ii port and send them to our Raspberry pie. Where we will use a software which is built using python and PyOBD package this software is to derive the data and showing them in the screen of our Raspberry Pie. In this software, we implemented our detection algorithm to detect dizziness, yawn, seatbelt and side mirror checking as well. Any abnormality that is described in the Evaluation Criteria section that is needed to processed by the Raspberry Pie will be sent to server. Then from the server our mobile app will derive the data and evaluate the data and give the decision upon this. Then the deduction of point will be done in the app and it will be shown to the owner with all the information we derived from the system. The user will also feel up some form according to his experience with the driver which will be helpful to evaluate the driver's performance. The app UI will provide the form to the user and then this input values will be sent to the app and then necessary evaluation will be done inside the app and then the evaluated data along with system derived data will be shown together to the user and as well stored in the server.



The Flow of Data



Flow of Data Inside Raspberry Pie

The tools Required in this project:

Software Tools:

Visual Studio Code: Visual Studio Code is an integrated development environment made by Microsoft for Windows, Linux and macOS

In our system we did all coding and testing by Visual Studio Code.

Visual Studio Code

Git hub: GitHub, Inc. is a provider of Internet hosting for software development and version control using Git. It offers the distributed version control and source code management functionality of Git, plus its own features.

In our system we were doing the coding parts in a laptop but we were running and testing it in a raspberry pi 4 computer. So, we code in laptop and pushed it to git-hub and pulled the code in raspberry pi and ran it.

Raspberry PI OS: Raspberry Pi OS (previously called Raspbian) is the official operating system for all models of the Raspberry Pi.

We ported this OS to the raspberry pi computer for a better GUI and hardware support.

Python: Python is an interpreted high-level general-purpose programming language.

We did our whole backend coding part using python. We used different python packages for example, opencv, obd, pyserial etc.

OpenCV: OpenCV is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez.

We used this to implement dizziness and yawn detection.

Flutter: Flutter is an open-source UI software development kit created by Google. It is used to develop cross platform applications for Android, iOS, Linux, Mac, Windows, Google Fuchsia, and the web from a single codebase.

We used flutter to develop the cross platform mobile app for the Vehicle owner.

Hardware Tools:

ELM327 Cable:

An ELM327 cable is an OBD2 diagnostic scan tool which connects to the On-Board Diagnostic (OBD) plug in your car. The 'language' spoken is broadly called OBD2.

In this System we are collecting the data from the car by this cable.

Raspberry PI 4 Computer Model B: Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation in association with Broadcom.

We used this computer to manipulate the collected data and also send required data to the server.

Testing Tool:

Raspberry PI Display: The Raspberry Pi Touch Display is an LCD display which connects to the Raspberry Pi through the DSI connector. In some situations, it allows for the use of both the HDMI and LCD displays at the same time (this requires software support).

We connected this display to the raspberry pi computer to test the code and get a better visual experience of what was happening.

Impact of the project on external factors:

Our project is conducted totally privately and on private property. There is no harm to anyone while conducting, testing and executing the project. It does not go against any rule of the government or any concerned organizations or any company policy. So, there is no legal constraint while carrying out the project.

Our Project's Effect:

Economy:

In future by using our app of "driver evolution and car safety" it will affect the economy because car owner can evaluate a driver and ensure the safety of the car. Therefore, it can help to reduce car damage. This will save the money of owner. Thus, cars will be more sustainable. As a result, the sales of the car will be decreased and importation of cars will be reduced which will help our economy grow.

Environment:

As we are providing the oil level report to the owner, the owner will be aware of the oil consumption of the car. If any issue happens, and the car consumes more oil that means releases more CO_2 to the environment. The owner can right away solve the problem and reduce the release of the extra CO_2.

Society/community:

Car safety impact a lot in our society because Motor vehicle crashes are the leading cause of death for people age 5 - 34. So, by using our app "driver evolution and car safety" is most effective way to save lives and reduce injuries in crashes. A vehicle's safety depends on several factors, which is already included in our app. So, people can easily relate and it will be helpful for them in several ways. This will certainly save some young lives from dying and they can leave an impact in the society.

Health and Safety:

As we discussed in the introduction part, 6,686 people lost their lives and 8,600 were injured in a total of 4,891 road accidents in 2020. This is a big number. Our project can help ensuring the owner about his driver's performance, car's diagnostic report. Therefore, the owner can change or confront the driver when he makes any error. This will make the driver more aware and careful. As a result, the accident rate must reduce.

Role of Individuals:

Role of Individuals So Far:

Md Shafaqul Islam Hamim:

I) UI Design.
II) System Design.
III) Dizziness and yawn Detection Algorithm.
IV) Evaluation Parameter Selection.
V) Seatbelt Detection Algorithm.

Muid Ahmed Alvi:

I) App Development.
II) Dizziness and yawn Detection Algorithm.
III) Evaluation Parameter Selection.
IV) Seatbelt Detection Algorithm.
V) PyOBD.

Shahida Akter Chowdhury:

I) Research on Existing Papers.
II) Project Report Contribution.
III) Evaluation Parameter Selection.

Functional Requirements:

i) Our system should be able to detect dizziness and yawn.
ii) Should detect seatbelt.
iii) Should provide information from the car to the user.
iv) Should Display all the evaluated data to the user.
v) User should be able to receive real time notification.
vi) User should be able to have more than one driver profiles.
vii) User can register multiple cars.
viii) User should be able to fill up monthly user experience index.

Division of Work:

Md Shafaqul Islam Hamim:

Frontend programming, Design, Detection Algorithms, Coding problems.

Muid Ahmed Alvi:

Backend programming, Car data derivation, Raspberry pie Connection.

Shahida Akter Chowdhury:

Planning, Article Writing, management.