Auto Shield Protection using IoT

Sai Prateek Atluri
School of Engineering and Technology
University of Washington
Tacoma, United States
asp109@uw.edu

Ankit Kadian
School of Engineering and Technology
University of Washington
Tacoma, United States
ankit24@uw.edu

***Abstract*—This project is about protecting cars from theft and monitoring the vehicle in real time. In this project we designed a model that can be installed on car dashboard, which is like advance driving assistant but also with increased features like Face authentication to start and drive the car. In addition to this the user will also see the live location, over speeding details, and it will monitor the Consciousness of the driver and if the driver is feeling sleepy it will make an alert to avoid accident chances. Furthermore, notification via email is also sent if an intruder tries to access the car.**

***Keywords*—** Node red, flow, Influx DB, Azure, Grafana, IoT sensors, dashboard, raspberry pi, Internet of things, Custom Vision.

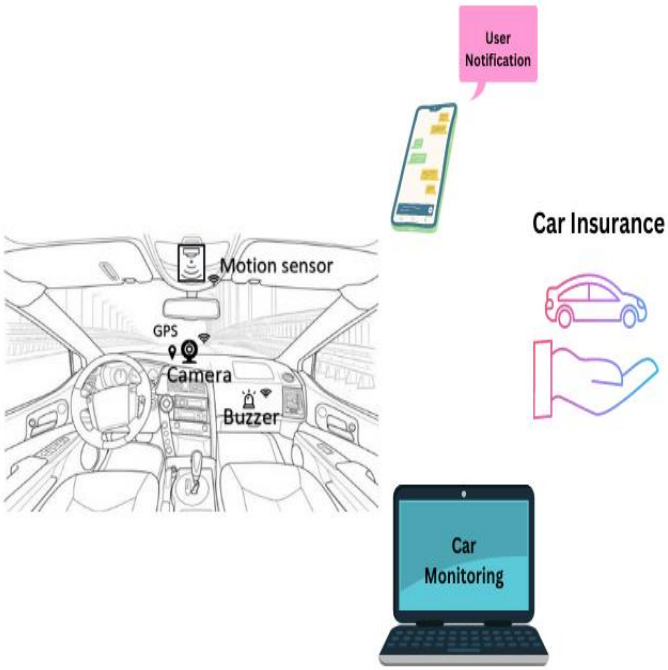
I. Introduction to Auto Shield

In this project we are solving the issue of car safety and in addition to that the monitoring of the driver and the car is also addressed. This model uses IoT sensors which are supported by grove pi and the grove pi board is mounted on raspberry pi board. Node Red is the main programming tool, which is flow based development tool for visual programming. This device captures the image of the driver and then confirm it that the driver is authorised or not if driver is authorised then the car will start if driver is not authorised then car alarm will go off and it will also send an email notification to the owner of the car.

The camera module is activated when someone comes in front of the dashboard of the car it is activated by the ultrasonic sensor someone enters the car, after that image is forwarded to Microsoft Azure custom vision model where already a trained model of the authenticated users verifies that user is already in the database or not. Since the user is authenticated then the camera module is activated, and it monitors the consciousness of the driver every 2 seconds and if the driver feels dizziness it will quickly make an alert so that the driver doesn't sleep and reduces the chance of accidents. It also stores the speed data of the car which is generated by the angle rotatory sensor and if the driver over speeds the speed limits, then there is an algorithm which will automatically applies traffic penalties to the licence of the driver. If the driver maintains a good driving style, then there will be no penalty on the driving licence in case if there are any driving penalties on the driving licence then the insurance charges will rise according to the driving style so if someone drives under speed limits then insurance charges can be reduced as there is less risk of accident.

II. System Architecture and Design

Below is a layout of the environment of the car dashboard where system is to be installed.



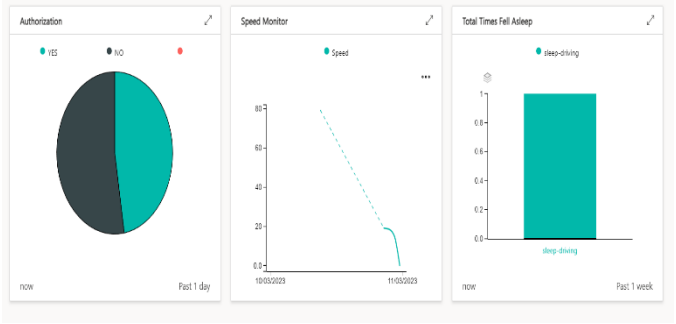
First a distance node is created in node red which calculates the distance using the ultrasonic distance sensor which is connected to digital pin of grove pi. when distance is in some range the camera module is activated the camera module is connected to Raspberry Pi the camera module takes the picture and the image is saved in base 64 format and then image is also checked with Azure custom vision model if the user is already in the Azure custom vision model then it will authenticate the most recent

image is always shown on the Grafana dashboard, once the user is authenticated the camera node will be activated and then it will click pictures every 2 seconds and check with the Azure custom vision model that the driver is active or not if he is feeling sleepy or dizziness then it will make an alert so that it can wake the driver. If the driver is feeling sleepy then those images are sent to influx database. If the user is not authenticated, then the car alarm will go off. In node red there is an email service node which Sends the email using smtp server so if the user is not authenticated or some intruder tries to get into the car in this case the owner of the car will be notified via email that someone is trying to get into the car. This node is connected to the result node of Azure custom vision model. There is one new node for speed monitoring of the car which is using the angle rotatory sensor which is connected to the analogue pin of the grove pi, here the speed of the car is observed after every 5 minutes and a threshold is set in the city area if the average off the speed recorded is higher than the threshold then penalty will be applied to the license of the driver. As penalty is applied the insurance charges off the car will increase is over speeding car is more prone to accident. The penalty is also applied if the driver is feeling dizziness since alert is sent to the driver so the penalty for this is very small as compared to the penalty of over speeding. The speeding data is stored in influx database and the penalty is calculated according to the speed limit threshold, the threshold for the speed limit is set to 40 mph. If the driver feels dizziness, then those images are also saved in the influx database, if driver is active then that data is not stored. the number of attempts may to authorise for the news of the car are sent to the Azure cloud along with the number of times driver felt sleepy or dizziness and the Car monitoring data is also saved in the Azure cloud and displayed only Azure dashboard.

The location displayed on the map is determined with IP addresses, with IP addresses we get the latitude and longitude of the location and then wrote a function to convert that latitude and longitude data to the location and pass it to the world map and display on the world map.

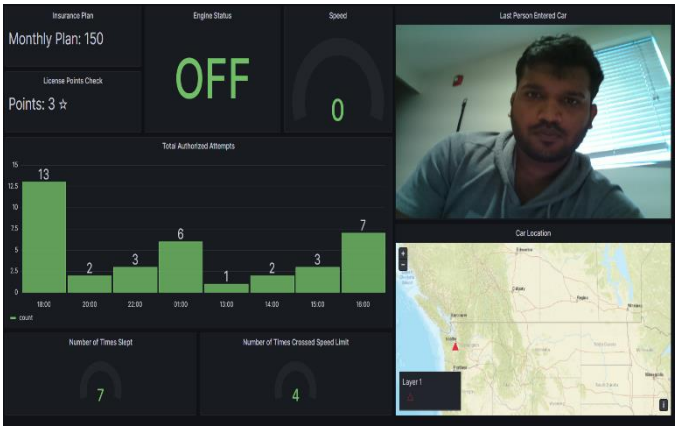
The Grafana dashboard shows how many times the driver felt dizziness number of times the driver cross the speed limit, the engine status that is on or off also it shows the number of attempts when someone tried to enter the car also shows the instant speed of the car it also shows the insurance plan according to the driving habits of the driver there is also one more feature which shows the live location of the car every second on the map. the Grafana dashboard also indicates the penalty points that has been applied to the licence of the driver and the owner of the car can you see who the last person is entered the car below is snapshot off the graph on a dashboard showing multiple information about the car and the driver the speed data is saved in the influx database.

Below is the Azure IoT dashboard, which shows their charts, the first one shows the Authentication attempts vs failed attempts in pie chart, the next one shows as line graph for the speed of the car and the last is the bar graph showing the number of times dizziness detected by the camera while driving.



The next image displays the Grafana dashboard which retrieves all the data from influx database which shows insurance plan, Engine status, License penalty points, speed of the car, number of the attempts made each hour to start and use the car, number of times over speeding recorded, number of times driver felt dizziness, live location of the car on map and the image of the person who last entered the car.

The Grafana dashboard collects the data from influx database and the data in influx is stored according to time, so we access the data according to the time and then Grafana can access the data according to the timestamps, the data can be overwritten by us as we are overwriting the image data and showing the data last image which camera clicked when someone attempted to enter the car.

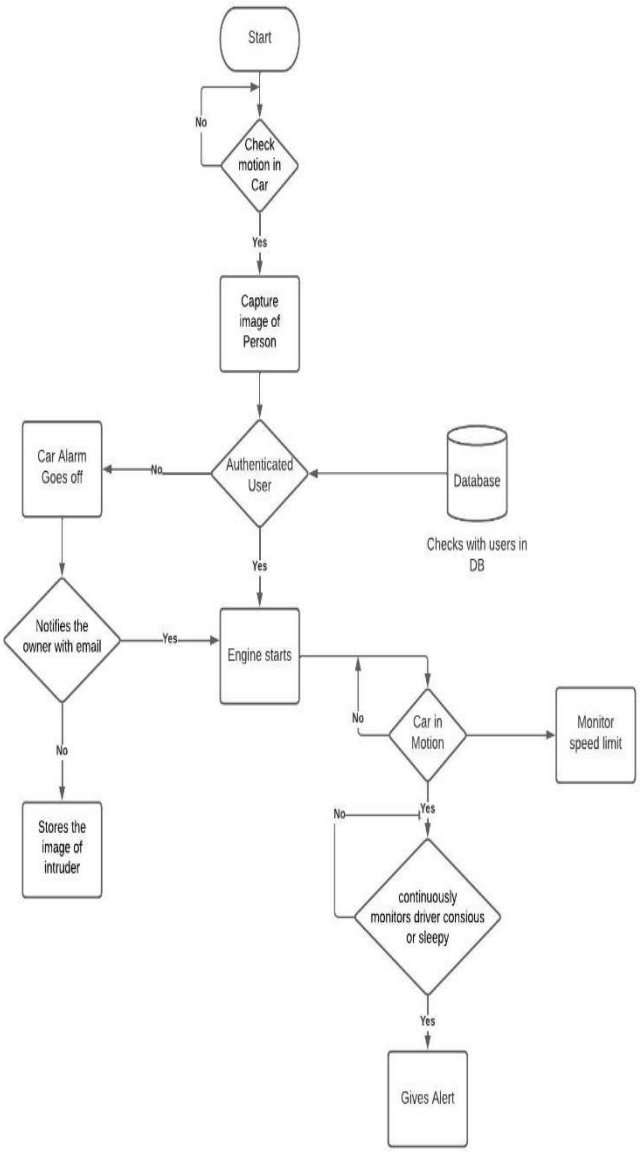


III. Discussion

The setup of the camera node was the first thing in this node flow that required a little research on internet, and it took time to figure out how to store images. The main issue was to train the custom vision model for Dizziness detection, trained many models with our dataset and few images from internet, then it worked well for us. The issue while working with Grafana was to display the image of the driver, so first we converted the image to base 64 and then in Grafana will tribe the image in base 64 and converted back to the image format this is how we displayed the image on Grafana dashboard. Meanwhile on Microsoft Azure dashboard the details of the attempts of authentication in the car is provided, in addition to this The

monitoring data is also sent to the Azure cloud and displayed on the dashboard. The image data and the monitoring data is also stored locally on the influx database.

The below is a flow chart of working of the main functions and decisions of the IoT device employed in Car.



IV. Conclusion

The performance of this IoT prototype is working fine, the authentication part and then after authentication the monitoring of the speed and the driver is activated. It is very helpful in deciding the insurance charges for the user, this can prove very useful for insurance companies. On the user end it also ensures that the car is safe and not stolen by anyone, as the dashboard shows the image of the person who entered the car last time and also provides a notification via email to the user that someone is trying to enter their car. This gives a piece of mind to the owner of the car. The data stored on the influx database is used to calculate the penalties did by the driver to decide their insurance plan and the over speeding penalties can be stamped on their licence.

V. Future Work

The algorithm for penalty and insurance can be improvised by recording the speed limits from maps of different places so that we can monitor the speed everywhere, which will help us to calculate the insurance amount very precisely. In this case the speed limit will automatically change in the algorithm according to the location of the car. This sleep monitor can also be improved for fatigue detection by using heart rate monitor and oximeter to detect fatigue, the heart rate monitor sensor and oximeter can be mounted on the steering of the vehicle. This will help in very precise monitoring of the driver. The GPS module can also be used to determine the location from the satellite and then show the location on the world map.

References

[1] M. Lekić and G. Gardašević, "IoT sensor integration to Node-RED platform," *2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH)*, East Sarajevo, Bosnia and Herzegovina, 2018, pp. 1-5, doi: 10.1109/INFOTEH.2018.8345544.J.

[2] M. S. Bin Mohd Nazri, T. Long Alif Faiqal Bin Tengku Long Gaafar, H. Sofian and A. A. Bakar Sajak, "IoT Parking Apps with Car Plate Recognition for Smart City using Node Red," 2020 11th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, 2020, pp. 324-330, doi: 10.1109/ICICS49469.2020.239511.

[3] J. A. C. Yatan, A. K. Mahamad, S. Saon, M. A. B. Ahmadon and S. Yamaguchi, "Earthquake monitoring system," 2017 IEEE International Symposium on Consumer Electronics (ISCE), Kuala Lumpur, Malaysia, 2017, pp. 61-64, doi: 10.1109/ISCE.2017.8355548.

[4] Salama AbdELminaam D, Almansori AM, Taha M, Badr E (2020) A deep facial recognition system using computational intelligent algorithms. https://doi.org/10.1371/journal.pone.0242269