PES UNIVERSITY

PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
100-ft Ring Road, Bengaluru – 560 085, Karnataka, India

A project report on:

Cooperative Intelligence Mobility

Done by:

Anuvrat Singhal	PES1201700299
Gaurav Jain	PES1201701834
Anirudh Krishnan	PES1201700334
Rishab K	PES1201700304
Hita Anil Kumar	PES1201701282
Sachin S	PES1201701006

In association with

NOKIA University collaboration 2020-21

CERTIFICATE

This is to certify that the Report entitled

'Cooperative Intelligence Mobility'

Is a bonafide work carried out by

Gaurav Jain (PES1201701834)

Anuvrat Singhal (PES1201700299)

Anirudh Krishnan (PES1201700334)

Rishab K (PES1201700304)

Hita Anil Kumar(PES1201701282)

Sachin S (PES1201701006)

In partial fulfilment for the completion of 6th semester Nokia project in the Program of Study of B.Tech in Electronics and Communication Engineering under rules and regulations of PES University, Bangalore during the period Jan – May 2020. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report. The report has been approved as it satisfies the 6th-semester academic requirements in respect of Nokia project work.

Signature with date & Seal
Guide & Chairperson
(Prof. Rajasekar)

Name of the examiners:

1. Mr.Kaustav Saha
2. Mr.Rajat Duggal

TABLE OF CONTENTS

ABSTRACT

The frequency of traffic congestion becomes high in India as the density of vehicles in a particular region increases. At traffic signals, people waste a lot of time particularly during the peak hours of the day and also sometimes due to bad road conditions like potholes, open drains or on-going construction.

Thus an idea of sharing real-time information about traffic and road conditions among road users is being proposed. The theme is to determine the total traffic and traffic density on each side of the road by calculating the number of vehicles at the traffic signal zone at regular intervals of time by using Image processing techniques. If a user is in the vicinity of this juncture, he/she is then notified about the status of nearby signals(whether there is congestion or not) and if the traffic density is high, a suggestion to reroute is also given to the user.

By analyzing real-time traffic data collected from the roadside infrastructure and the vehicles on the road, Intelligent Transportation Systems would be able to timely warn road users in the vicinity about bad road conditions, traffic blocks and impending safety-critical situations ahead, increasing road safety and traffic efficiency.

ACKNOWLEDGEMENTS

We would like to take this opportunity to express our profound gratitude to all those people who were directly or indirectly involved in the completion of this project. We thank each and everyone who encouraged us in every possible way.

We would like to thank **Prof. Jawahar,** Chancellor, PES University, **Dr Anuradha M**, HOD, ECE department, PES University for letting us be a part of this prestigious institution and letting us explore our abilities to the fullest.

We would like to express our heartfelt gratitude towards our project guide **Mr.Kaustuv Saha** for the constant guidance, valuable knowledge and experience.

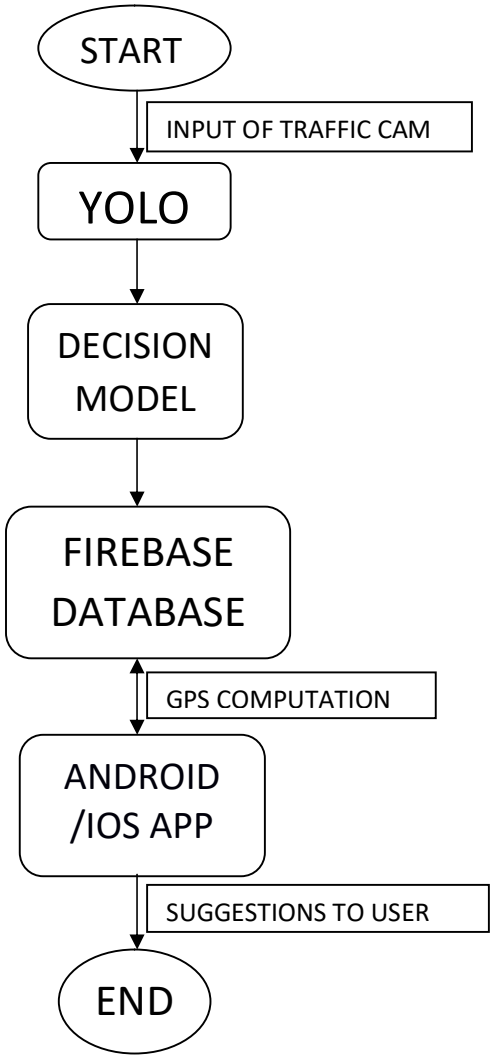
We would like to thank our course coordinator **Prof.Rajasekar**, for introducing this course to us. And helping us out whenever possible.

INTRODUCTION

Digital technologies have been developing rapidly over the past few years and the scope of introducing them in the field of transport and road safety is being realised. Cooperative Intelligence Mobility focuses on digital technologies providing data analytics for road users, to ease traffic density on roads and thereby making it a much more efficient solution to tackle the issue of road congestion

Our project 'Cooperative Intelligence Mobility' focuses on providing real-time data about the number of vehicles and traffic density to the user and a suggestion to take an alternate path if the conditions are bad.

DESIGN


START
INPUT OF TRAFFIC CAM
YOLO
DECISION MODEL
FIREBASE DATABASE
GPS COMPUTATION
ANDROID /IOS APP
SUGGESTIONS TO USER
END

IMPLEMENTATION

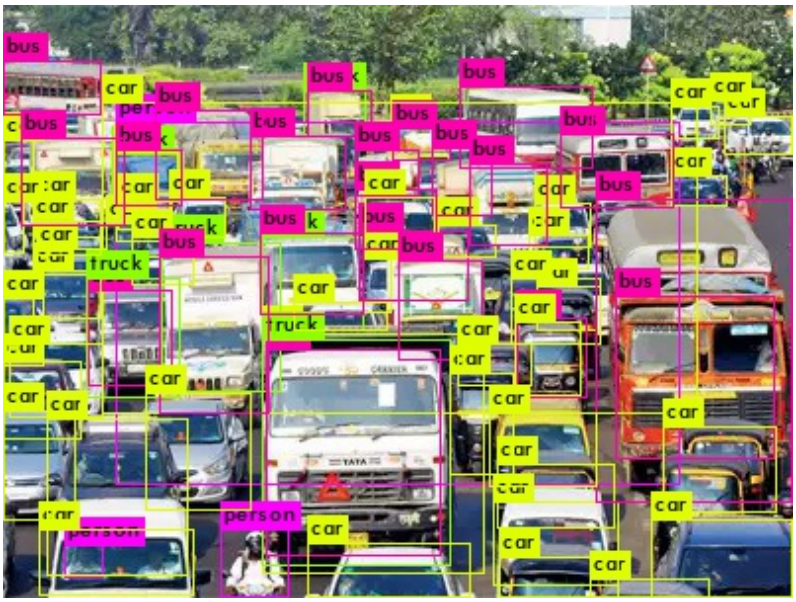
1. YOLO:

YOLO (You Only Look Once) is a real-time object detection system. YOLO architecture is based on FCNN(Fully Conventional Neural Network). We feed the system with an image and then it passes the image of (nxn) size through the FCNN and an output image of (mxm) size is predicted. Then what the architecture does is it splits the input image into multiple grids. Each grid cell is predicting bounding boxes and also the confident scores for those bounding boxes. These confident scores reflect how confident the model is, that the box contains an object and how accurate it thinks the box is, that it predicts. If no object exists in the cell, the confidence scores would be zero. Otherwise the confidence score will be equal to the Intersection over union(IOU) between the prediction box and the ground truth box.

For our implementation, we assume that we have access to the traffic cams at the signals. With this live video feed, we capture a frame every few seconds and send this image to be processed via YOLO. The output is reformatted to give us the total number of cars, bikes, trucks and buses that have been realised.

Another advantage of using YOLO over other methods is that it also gives us a probability of accuracy for each prediction that it makes. Here we were able to implement a basic filter that only considers an object that has its prediction accuracy probability over a threshold limit (we used 40 %). This is so that we could find a balance between speed and accuracy with speed being important to handle real time data such as our case.

The output is the total number of vehicles of each type - bus, truck, car and bike and this is sent to the decision model for further analysis.



2. Decision model:

The algorithm does 3 operations from the data it has collected:

1. In a real world scenario, the physical space taken by one bike is not the same as the physical space taken up by one car or one truck. Hence it wouldn't be accurate to measure solely based on the number of vehicles. Therefore we create a new parameter called traffic density. Here the size of the vehicle captured is taken into account and the value obtained from this variable is taken for further classification.

2. The traffic density parameter can vary with a huge range according to some of our test cases. Hence we used the concept of weighted average and reduced the range to values from 0 to 20 so that comparison and classification becomes easier.

3. It classifies the junction into three categories, namely :
    - No Congestion: if there are very few or no cars waiting at the signal, a '**LOW'** congestion rating is generated.
    - Moderately Congested: if the number of vehicles waiting crosses a threshold value but is still below the saturation value, a '**MEDIUM'** congestion rating is generated.
    - Highly Congested: If the number of vehicles waiting crosses the saturation value, then the '**HIGH'** congestion rating is generated.

```
============= RESTART: G:\darknet-master\build\darknet\x64\alpha.py ============
Signal 1

The three values printed below under each catogory of vehicle are as follows:
 1.Total number of vehicles detected.
 2.The number of valid cars after passing through the decision model.
 3.The value after adding the weights.


car 41 35 35
bus 0 0 0
bike 0 0 0
truck 4 2 3
Total traffic 37
Total traffic density 8.636363636363635
Rating MEDIUM



Signal 2

The three values printed below under each catogory of vehicle are as follows:
 1.Total number of vehicles detected.
 2.The number of valid cars after passing through the decision model.
 3.The value after adding the weights.


car 2 0 0
bus 1 1 2
bike 39 30 15
truck 2 2 3
Total traffic 33
Total traffic density 4.545454545454545
Rating LOW
>>>
```

3.Firebase Real-time Database

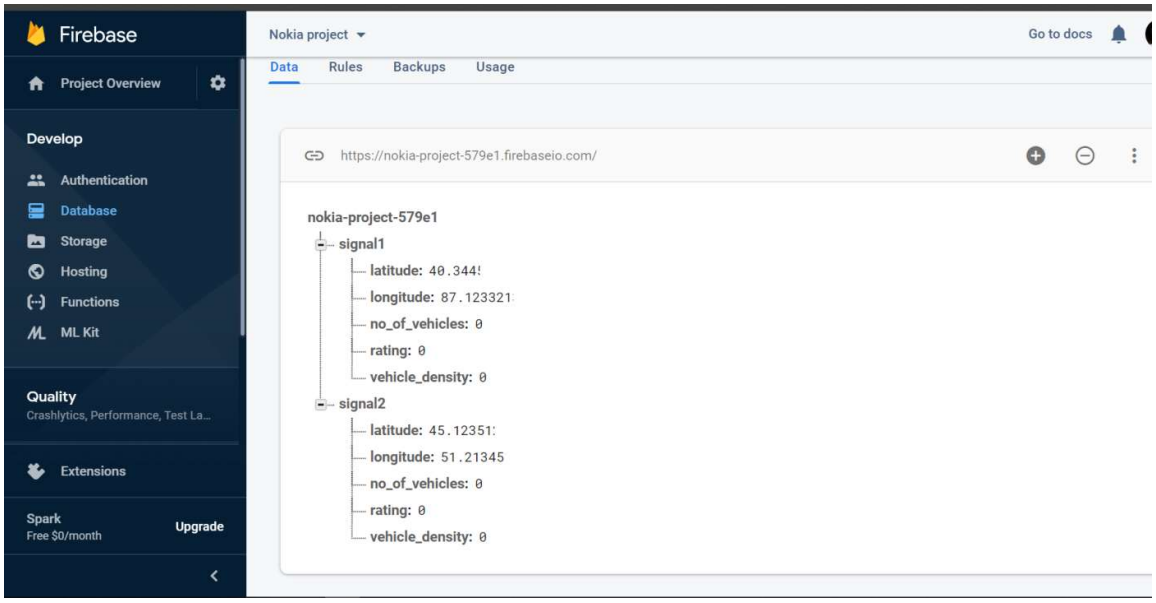
The Firebase Real-time Database is a cloud-hosted database. Data is stored as JSON and synchronized in real-time to every connected client. The advantage of using Firebase is its cross-platform capabilities. All users from IOS, Android and JavaScript SDKs, all access the same instance of the database and automatically receive updates with the newest data.

In our model, the database stores certain information about each signal like its latitude, longitude, no of vehicles currently at the signal, the density of vehicles at the signal and also the congestion level.

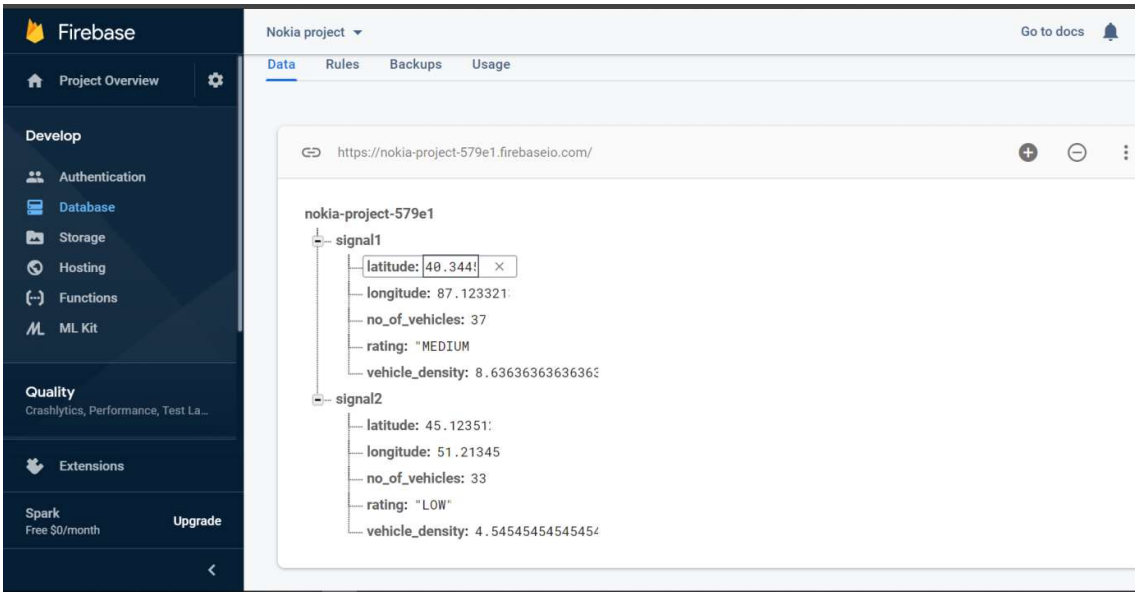
Here since we need not change the gps coordinates of the signal (latitude and longitude), we only update the other parameters periodically using a simple python script with values that have been processed through YOLO.

The android app needs to retrieve information from the database and this is done using Java from the app end and each instance of a signal is saved as an object in java.

Before updating the database:



After updating the database:



4. GPS Parameter

Another aspect that was looked into was how the app is able to retrieve data only for signals in its vicinity. Every time a new signal is added to the database, its latitude and longitude is manually entered and stored. The app handles the computation to check whether it's in the vicinity of a signal by taking its own gps coordinates and then comparing it to that of a signal. The distance between the two points is computed using basic mathematics and if it's below a certain threshold value then the signal data for that particular signal is displayed to that user.

5. Android App

Features of the app:

- Retrieve traffic density, signal strength and other information from the database.
- Detect the current location of the phone and compare the location with GPS coordinates of the signal retrieved from the database.
- Reroute if necessary.

Working of the app:

- Retrieval of data from the database:

  The app is being developed using Android Studio and the interface extracts the GPS coordinates from the user's phone and it is stored as a variable in the app. The current status of the signal's having the feed is send, which is evaluated by the backend systems. Then it’s connected to Firebase and it retrieves information like the GPS coordinates, Traffic density, Signal rating and the number of vehicles present directly from the Firebase database for every signal junction and displays it. This firebase is constantly updated every 30 seconds. This data is then sent to the app from firebase and the coordinates is transformed into perceivable distances.

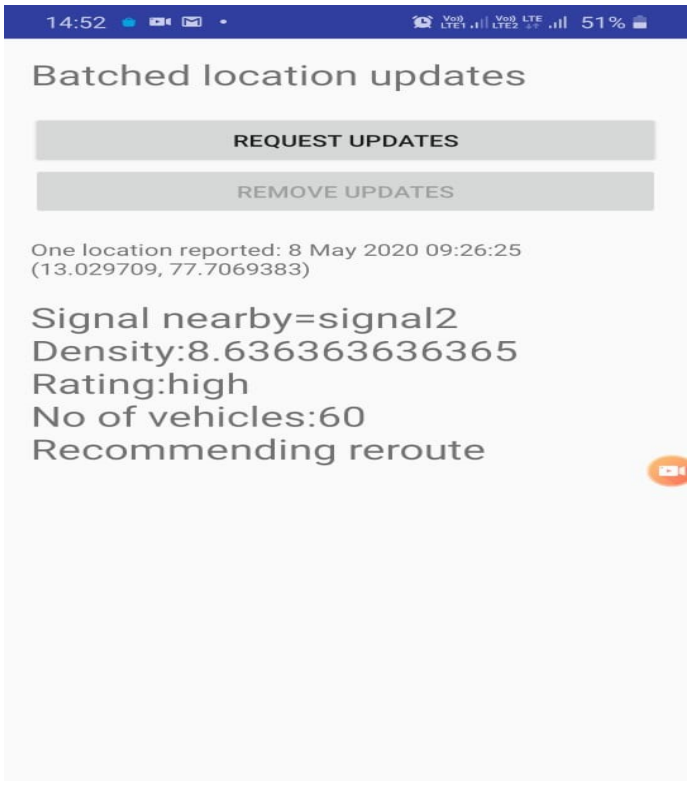
- Detecting the current location:

The app detects the current location of the phone and compares it with the GPS coordinates of the various traffic signals received from the database. The current status of the signal's feed is send, which is evaluated by the backend systems. Using these coordinates, it identifies the signal closest to its current location. The distance of each signal from the user is quantified and a threshold is set for the checking whether the signal is within the user's vicinity. We then display the information of the traffic signal thus identified.

- Rerouting:

A notification about the signal that the user is approaching is popped up and based on the traffic density and the signal rating, the user can make a decision to reroute by clicking on the "Reroute" button provided. Doing so will directly take him to Google Maps, where he can decide the alternate route to follow and make adjustments accordingly.

Interface of the App:



.

FUTURE SCOPE

For the ease of our project we have limited ourselves in a few areas which can further be dealt with to make the model more robust:

- Instead of just using 2 signals in the database, more signals' data can be easily appended and used.
- Decision model works on basic mathematical functions and can be made more complex to yield better output.
- We are limited on accuracy of prediction depending on the way the traffic camera is placed at the signal.
- The app interface can be made more user friendly instead of just a basic layout.
- Rather than just suggesting the user to reroute, if we can automatically provide an alternative route to take it would improve the application aspect of our model.
- Since we don't have access to live cam feed from signals, we are directly feeding images to YOLO.

BIBLIOGRAPHY

1) https://pjreddie.com/darknet/yolo/
2) https://towardsdatascience.com/yolo-you-only-look-once-real-time-object-detection-explained-492dc9230006
3) https://etsc.eu/wp-content/uploads/ETSC-Briefing-on-Cooperative-Intelligent-Transport-Systems-C-ITS.pdf
4) https://firebase.google.com/docs/database
5) https://developer.android.com/docs
6) https://developers.google.com/maps/documentation