A Project Report on

Traffix : Efficient Traffic Control using IoT

Submitted by

Reuel Amin (Roll no: 01)

Alarik Correa (Roll no: 09)

Renoy Dsouza (Roll no: 13)

Elvina Fernandes (Roll no: 19)

in partial fulfillment for the award of the degree

BACHELOR OF ENGINEERING *in*

Computer Engineering

Under the Guidance of

Prof. Prachiti Pimple

St. Francis Institute of Technology, Mumbai

University of Mumbai

2023 - 2024

CERTIFICATE

This is to certify that **Reuel Amin, Alarik Correa, Renoy Dsouza, Elvina Fernandes** are the bonafide students of St. Francis Institute of Technology, Mumbai. They have successfully carried out the project titled ***"Traffix: Efficient traffic control using IOT"*** in partial fulfillment of the requirement of B.E. Degree in Computer Engineering of Mumbai University during the academic year 2023-2024. The work has not been presented elsewhere for the award of any other degree or diploma prior to this.

(Prof. Prachiti Pimple)

(Dr. Kavita Sonawane)

CMPN HOD

(Dr. Sincy George)

Principal

Project Report Approval for B.E.

This project entitled ***'Traffix: Efficient traffic control using IOT'*** by **Reuel Amin, Alarik Correa, Renoy Dsouza, Elvina Fernandes** is approved for the degree of Bachelor of Engineering in Computer Engineering from University of Mumbai.

Examiners

1. ------------------- 7/5/24

2. ------------------- 7-5-24

Date: 07/05/2024

Place: MUMBAI

ACKNOWLEDGEMENT

We had a great experience working on this project as a group and we got to learn new skills which have helped us inculcate some genuine knowledge about various domains. However, it wouldn't have been possible without the kind support and help of many individuals. We would like to extend our sincere thanks to all of them. We are highly indebted to the Director Bro.Shantilal Kujur, the Principal Dr. Sincy George, the HOD Dr. Kavita Sonawane, the teaching and non-teaching staff for their guidance and constant supervision. Special thanks to Prof. Prachiti Pimple for always guiding and providing us with every possible means of information throughout the project. Their support truly helped us stay motivated during the entire course of preparation. Lastly we would like to thank the entire department of CMPN for providing facilities and a conducive environment in making this project.

Signatures of all the students in the group

(Reuel Amin)

(Alarik Correa)

(Renoy Dsouza)

(Elvina Fernandes)

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included; we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in this submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Signatures of all the students in the group

(Reuel Amin)

(Alarik Correa)

(Renoy Dsouza)

(Elvina Fernandes)

Abstract

Our project intends to reduce traffic congestion brought about by antiquated, inefficient traffic management technologies that operate on a predetermined countdown. Traditional systems assign timings without taking into account the actual traffic density on a given road, which results in significant red light delays. Our project provides a solution that ensures that traffic light durations are based on real-time traffic conditions, instead of working on static allotted timings, allowing for proper management of resources and time. Our project is based on calculating the current traffic density, which is derived using a combination of image processing techniques and ultrasonic sensors.

A Raspberry Pi at junctions is responsible for allocating the durations for the traffic signals. It achieves this by input from cameras and sensors to manage the signal lights at intersections. Additionally, this processed data from Raspberry Pi and is uploaded to a database called SqlAlchemy. This data can be accessed and utilized to periodically check traffic flow intervals. The values saved in the cloud will also be helpful in anticipating the density of traffic in the event of system failure due to hardware failure or extreme difference in values from the camera and the ultrasonic sensors.

The smooth passage of vehicles is made possible by this traffic management system, which also incorporates a fail-safe system that will be helpful in unforeseen events.

Keywords: IoT, Image Processing, Raspberry Pi, Machine Learning, SqlAlchemy, Traffic Congestion, Ultrasonic Sensors.

Contents

List of Figures

List of Abbreviations

RSU	Road Side Unit
GPS	Global Positioning System
RFID	Radio Frequency Identification Device
ANFIS	Adaptive Neuro Fuzzy Inference System

List of Tables

Chapter 1

Introduction

India’s road network has grown at an annual rate of 4% since 1951 but the number of vehicles has increased nearly 11%, choking roads and increasing pollution. India has the second largest road network in the world with 4.2 million km. Interestingly, even though the increase in road length has kept pace with the rising population, it has not managed to keep pace with the increasing number of motor vehicles. Delhi has the highest number of vehicles at 6.3 million. Ahmedabad and Chennai also feature prominently with 1.7 million and 2.9 million, respectively. Roads are essential for the growth of a nation, with India being a developing country our road network is of great importance. With the growth in our economy and population, our road networks cannot grow anymore without drastic changes to the already existing infrastructure. Our project proposes a solution to manage the ever increasing traffic and congestion without drastic changes to the underlying infrastructure.

1.1 Description

Traffix offers a solution to ever increasing traffic and congestion problems without drastic changes to already existing infrastructure. Using inputs from image processing systems and ultrasonic sensors, the green time at every traffic signal can be allocated dynamically according to the density of traffic in each lane.

1.2 Problem Formulation

Traffic with poor regulation leads to traffic congestion, many of which are linked to the signal shutting down too quickly leading to agitation of people stuck in traffic causing breaking of signals which increases the risk of accidents ultimately leading to junction congestion.

1.3 Motivation

At some point we all have experienced being stuck in a traffic where you are at a four cross junction with vehicles in front of yours and all vehicles in your lane are stuck due a red light while the light turns green for all the lanes while there are close to none or very little vehicles passing through those lanes. And when the light turns green in your lane due the amount of traffic in front of you, you are unable to pass as the light turns red again making you have to wait another turn of this just to pass. This leads to people breaking the signals and getting in accidents ultimately causing even more traffic. This also causes loss of precious time and energy for the people stuck in such situations.

1.4 Proposed solution

Traffix is a solution to traffic congestion and effective traffic management. It takes inputs from cameras for image processing and ultrasonic sensors for object detection. The main task for the system will be to dynamically allocate time to traffic signals detecting the traffic density on each lane using Raspberry Pi. In scenarios of hardware failure or when the inputs differ drastically and continuously the system will rely on data that was previously processed by Raspberry Pi and stored on SqlAlchemy.

Some additional features of Traffix are:

1. <u>RFID for Emergency Vehicles :</u>

This IOT based technology focuses on providing a solution to emergency vehicles majorly ambulances during heavy traffics. The system works in such a way that the RFID scanner gets the information from a RFID tag attached to the ambulance. As soon as the scanner scans the tag it alerts the traffic signals in the path of the ambulance by sending a message to display on a LCD screen just before the signal. This way it provides a “green corridor”, that is the vehicles in the lane of the ambulance will progressively shift lanes providing a free road for the ambulance or any other emergency service vehicle without any hassle.

2. Database Support :

The processed data from Raspberry Pi and is uploaded to a database called SqlAlchemy. This data can be accessed and utilized to periodically check traffic flow intervals. The values saved in the cloud will also be helpful in anticipating the density of traffic in the event of system failure due to hardware failure or extreme difference in values from the camera and the ultrasonic sensors.

1.5 Scope of the project

The domain of this project falls under IoT and Machine learning. For now this project is limited to a four cross junction but in the future can also be applied to a three cross junction with more added features. The features can also be used as stand-alone features and applied to various parts.

Chapter 2

Review of Literature

Our main aim of this project is to reduce traffic congestion caused by inadequate, inefficient traditional traffic technology by implementing a dynamic cloud based traffic management system. Some of the literature papers that highly motivated and supported us through the project have been listed with its methodology, advantages and disadvantages.

Paper [1] speaks about eliminating the waiting period required for each vehicle in the traffic by providing them the information of the congestion of vehicles in that particular signal beforehand by using an RSU(Road Side Unit). Each vehicle will have an RFID tag and each traffic signal will have a RSU and an RFID scanner. As the vehicle passes through each signal the RFID scanner scans the tag on each vehicle and sends the data to a database system through RSU. The database will have the total number of vehicles on that particular signal. Now, every user will have an HTML5 based mobile app which will notify them of the traffic at every signal and the distance from that vehicle through GPS. It will also help estimate the speed required to pass that signal without having to wait in the red light signal causing an increase in the waiting time. The major disadvantages of this system are that even though it eliminates the waiting period for vehicles, any sudden changes in the traffic order caused by accidents or any similar situation could cause problems. This is because the database won't have information about the scenarios causing delays between the two signals as the RFID and RSU are only connected to the traffic signals. The driver will hence assume the previous information received as the correct one and continue to travel at the same speed. The solution we have established for this problem is attaching RFID tags only to emergency vehicles since every vehicle containing an RFID tag could cause a tremendous load on the database. We also have cameras which will help us perform image processing and gain real-time data providing accurate signals at every junction using object detection.

Methods	Advantages	Disadvantages
● Road Side Unit(RSU), RFID, GPS	● The RSU is attached to every traffic signal with a camera attached to it to capture the real time image of the signal with its timer. It sends the data to a server. ● Every vehicle will have an RFID unit. A scanner at the traffic signals will send the vehicle details to the server. ● The server will send data on a mobile app and will let the vehicle the distance and speed required to pass a signal without waiting using GPS	● Major drawbacks are that even though it allows vehicles to speed past green light signals without waiting, sudden changes in the traffic order could cause accidents because of the time at which the vehicle gets the speed and distance info.

Paper [2] speaks briefly about how the existing system which is manually controlled has drawbacks due to many underlying conditions and how these drawbacks could be resolved by implementing a model that uses image processing. The drawbacks of the existing system include the need of manpower to be physically present inorder to control and manage the flow of traffic by using traffic rules based hand gestures. The existing automated traffic system is based on static allotment of time to individual signals based on sensors and camera input. These drawbacks are resolved by using image processing along with various algorithms including Gaussian filtering, Canny edge detection and time allocation. It uses these algorithms to find the density of vehicles on each lane by eliminating various noises from the image captured and establishing edges around vehicles to get the approximate density of each lane. Based on this it allocates the time to each signal and hence the signal with the highest density of vehicles will have a shorter period of waiting time. The major disadvantages of this system are that while image processing must occur continuously it could cause a tremendous load on the device which could cause inaccurate results. This system also does not prioritize emergency vehicles and accidental situations which requires immediate action on reducing the traffic by dynamically switching the signal to green.

The solution we have established from this system is using image processing along with an additional cloud based system that is SqlAlchemy. In any case if there is inaccurate information delivered through image processing we can rely on SqlAlchemy which will statically allocate signal timings based on past data.

Methods	Advantages	Disadvantages
● Image processing techniques like Gaussian filtering, Canny edge detection, time allocation algorithm	● Capture images of each lane and a reference image of an empty road and convert them to RGB. ● Apply Gaussian filtering to reduce noise. ● Apply Canny edge detection to smooth the images and detect the edges. ● Compare the reference image and the analyzed image for density detection. ● Dynamically switching signal lights based on density.	● System may not consider emergency vehicles since it only considers lane density of vehicles. ● Image processing must occur continuously which can cause a tremendous load leading to delays.

Paper [3] speaks about providing a green corridor for emergency vehicles using RFID technology. The main purpose of this system is to dynamically update the traffic signals based on an oncoming emergency vehicle. It then accordingly shifts lanes based on the updates received from the RFID tag and can easily reach their destination without having to wait in the congested traffic. The major disadvantages of this system are that while a fixed timing structure of the signals is followed there could still be some delay in dynamically changing the signal lights based on the vehicle. Inaccurate scanner which could occur due to weather conditions or various other factors hinders the effective communication between the scanner and the vehicle.

The solution that we have established from this paper is to send input about the emergency vehicle based on the camera's input as well as cloud based data. This ensures that the data about the emergency vehicles is constantly delivered and updated to the next signal in case of a malfunction in the RFID system.

Methods	Advantages	Disadvantages
● Utilizing RFID technology	● RFID scanner retrieves the information from an RFID tag attached to an ambulance/emergency vehicle. ● Scanner alerts the traffic signal based on the input received from the scanner and turns the signal green. ● Green corridor is established as ambulances can progressively shift lanes. ● Traffic lights switch to green and have a message system which alerts the vehicles of an oncoming emergency vehicle.	● Fixed timing structure of signals may cause delays. ● Inaccurate scanner may hinder effective communication with smart signals

Paper [4] speaks about using an Adaptive Neuro Fuzzy Inference System (ANFIS) using inputs such as the waiting time and vehicle density, developed via the MATLAB Simulink environment to improve traffic conditions. The least-square method and backpropagation gradient descent techniques are used for training the ANFIS. The images captured using camera sensors connected to the Arduino UNO are sent to the cloud that is SqlAlchemy for further processing of the image. There are two separate models i.e. Daytime vehicle detection model and Night time vehicle detection model. The daytime vehicle detection model extracts the images received into blocks of pixels using optical flow and background subtraction. It calculates the centroid of the vehicle for the distance traveled and its velocity. Background subtraction is used for detecting stationary vehicles. It uses a blob analysis algorithm for detecting the number of vehicles moving using Simulink. Based on this the traffic density is obtained. The method used for night time vehicle detection is Ostu’s method of image thresholding. The image is segmented between foreground class and background class and the blob analysis algorithm detects the headlights of the vehicle as blobs and calculates the vehicle density by giving a total count which is divided by 2. The ANFIS is then provided with a set of rules and the learning model then dynamically allocates the signal timing based on the queue length and waiting time. The major disadvantage of this system is that the inputs given to the ANFIS are preprocessed so the the output are therefore not completely interpretable. The overall computational cost of this system is high too due to its complex structure.

The solution we can establish for this system is to take smaller sets of inputs for every processing; this could reduce the computational complexity.

Methods	**Advantages**	**Disadvantages**
● Adaptive Neuro Fuzzy Inference System. ● Least-square method and backpropagation gradient descent. ● Ostu's image thresholding method. ● Blob analysis algorithm.	● Cloud based real-time image storage using ThingSpeak. ● Both daytime and night time vehicle density detection. ● Matlab Simulink Environment for inputs.	● High computational cost. ● System is not completely interpretable due to large inputs and complex structure.

Paper [5] speaks about estimating the traffic congestion by using LabView and ThingSpeak. The system basically works in four stages. At the first stage various analog sensors and laser cameras are used to calculate the total number and types of vehicles on the road by shooting a video stream and using it frame by frame. At the second stage the data pre-processing occurs, the analog data received is digitized using the analog-to-digital converter with the help of Arduino. This data is then sent to a single-board computer such as the Raspberry Pi where the data preprocessing takes place. This is the third stage. At this stage the traffic congestion ratio is calculated based on the speed, length of the road, width of the road, distance of the vehicle from each other and other parameters. This data is then transmitted to the Cloud. This helps to reduce the system load. The thingspeak helps get a detailed view on the levels in traffic congestion. In the fourth stage the LabView gives a proper representation of the congestion occurring at that moment i.e. from the frame received from the video stream. The ThingSpeak then generates the statistics of the total number of vehicles along with the time, date it was present and the type of the vehicle. The drawbacks are it has high complexity, needs a lot of time for data pre-processing, cheaper alternatives to camera could lead to inaccurate results.

The solution is to use good quality sensors and cameras to get the highest possible accurate results. Applying ML techniques to the system could help save time in data pre-processing. ThingSpeak can be used as a backup by storing some past data in case the main system fails.

Methods	Advantages	Disadvantages
● Labview ● MATLAB ● ThingSpeak ● Data-Preprocessing ● Data-Visualization	● Low Cost ● Accurate representation of data ● Cloud based storage	● High Complexity ● Data pre-processing requires more time ● Use of multiple sensors and cheaper alternatives to camera could lead to inaccurate results

Paper [6] speaks about detecting parked vehicles, which involves identifying objects that are halted for an extended period of time. This algorithm begins by utilizing the Harris algorithm to detect corner points in a given environment. These corners are then categorized into two groups: static and dynamic. Dynamic corners are likely associated with moving objects like vehicles and pedestrians, while static corners can be found on stationary cars and background elements. The camera monitoring areas, typically situated at junctions, extend vertically. By disregarding the transverse dimension of corners within the lane, one can create a one - dimensional vector for each frame. The proposed system is versatile and can be employed for real-time operations as well as offline statistical analysis of lane occupancy. This system provides a framework for traffic monitoring and analysis, offering both real-time insights and the ability to conduct in-depth statistical studies of lane usage. The Disadvantage is that the system offers a pragmatic approach for detecting parked vehicles in street lanes, it has limitations related to vehicle discrimination, false positives, and the absence of detailed vehicle attributes. Its suitability depends on the specific requirements of the application and the trade-offs between simplicity and precision. This may not be suitable for applications requiring precise vehicle tracking or recognition.

Methods	Advantages	Disadvantages
- Video Analysis - Traffic Image Analysis - Parked Vehicle Detection - Traffic Planning	- To detect the presence of parked vehicles in a particular street lane - The use of corner points as clues helps enhance the accuracy of vehicle detection algorithms by leveraging the textured information within the space occupied by the car in an image.	- The change in the background can disrupt the system's ability to accurately recognize the environment. - It can also lead to misclassification or difficulties in identifying the scene correctly after the car has left.

Chapter 3

System Analysis

3.1 Functional Requirements

1. Real-time Traffic Monitoring:

The system should be able to collect and analyze real-time traffic data using sensors, cameras, or other IoT devices to determine the traffic density at each signal.

2. Dynamic Signal Control:

The system must be capable of dynamically adjusting the signal timings based on the current traffic conditions, prioritizing traffic flow optimization and minimizing congestion.

3. Emergency Vehicle Priority:

The system should have the functionality to recognize and prioritize emergency vehicles, allowing them to navigate through the traffic more efficiently.

4. Data Analysis and Reporting:

Incorporate data analysis capabilities to generate comprehensive reports on traffic patterns, congestion levels, and signal performance, facilitating informed decision-making for future optimizations.

5. SqlAlchemy Integration:

The system must integrate the SqlAlchemy database to facilitate efficient data collection, analysis, and visualization for dynamic traffic signal time allotment to use this processed data in times of hardware failure.

6. Compatibility with Existing Infrastructure:

Ensure that the system is compatible with the existing traffic signal infrastructure, allowing for seamless integration and minimal disruption during the implementation phase.

3.2 Non Functional Requirements

- **Performance Requirements:**

1. **Real-Time Responsiveness:** The system should respond to changes in traffic conditions in real-time, updating signal timings promptly and accurately.
2. **Traffic Throughput:** Ensure that the system can handle high traffic throughput and effectively manage traffic flow during peak hours and congested periods.
3. **Low Latency Communication:** Minimize communication latency between the traffic sensors, central control unit, and the traffic signals to ensure timely data transmission and signal adjustments.

- **Safety Requirements:**

1. **Emergency Vehicle Prioritization:** Provide a mechanism to prioritize the passage of emergency vehicles, ensuring their swift and safe movement through traffic intersections.
2. **Pedestrian Safety Features:** Implement dedicated pedestrian crossing times, clearly visible signals, and safe pedestrian paths to minimize the risk of accidents.

- **Security Requirements:**

1. **Access Control and Authentication:** Implement robust access control mechanisms and user authentication protocols to prevent unauthorized access to the traffic signal control system.
2. **System Integrity Checks:** Conduct regular integrity checks and audits to identify potential security vulnerabilities and mitigate any potential risks of cyber-attacks or unauthorized system manipulations.

- **Software Quality Requirements:**

1. **Reliability and Fault Tolerance:** Design the system to be highly reliable, with built-in fault tolerance mechanisms to ensure continuous operation even in the event of device failures or network disruptions.

2. **User-Friendly Interface:** Develop an intuitive and user-friendly interface for traffic management authorities to configure settings, monitor traffic conditions, and analyze system performance effectively.

3.3 Specific Requirements

Hardware Requirements:

Raspberry Pi 4B

Ultrasonic Sensors

Raspberry Pi Camera module

RFID Scanner

RFID Tags

LED/LCD Screen

Signal Lights

Software Requirements:

Operating System : Windows

Programming Language : Python

3.4 Use case and Description

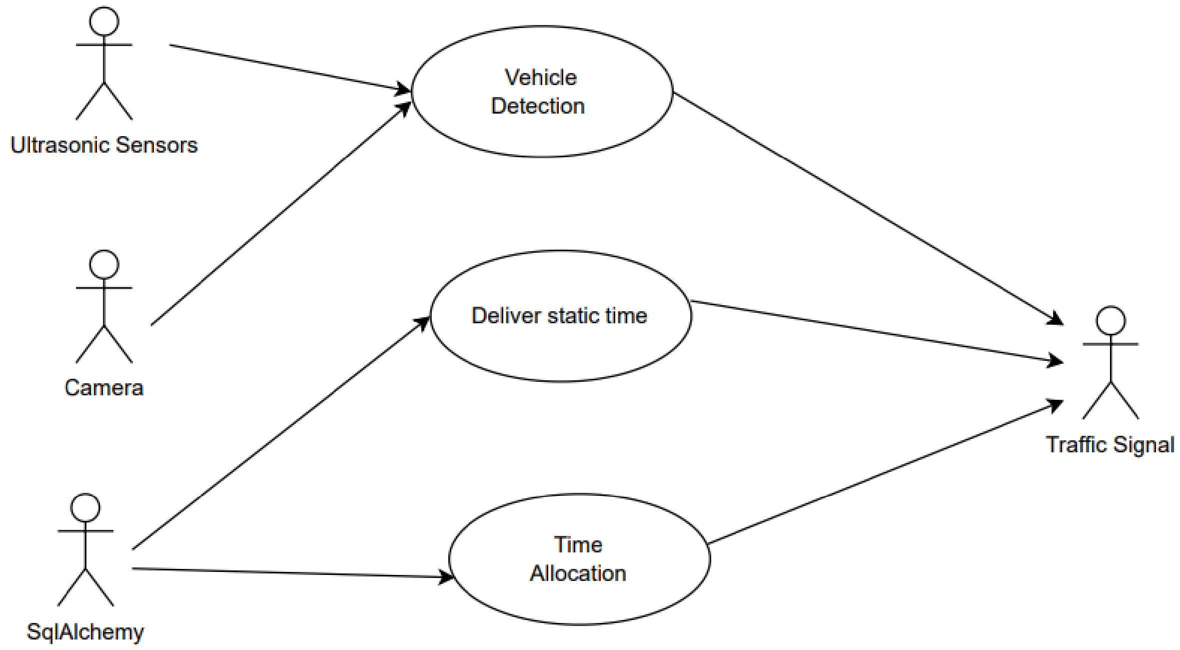


Figure 3.4.1: Use case diagram

Use case diagram description:

The above use case diagram (Figure 3.1) shows the high level function and scope of the system. The use case diagram indicates how the system works in response to various conditions and based on it indicates to the user the action it must perform. In the above use case diagram there are four actors. Actor Ultrasonic sensors and actor Cameras perform vehicle detection at different locations according to its placement. It then sends the data to the main traffic signal which is our fourth actor. It also receives static data from actor SqlAlchemy as its backup. The time allocation is sent to the traffic signal which is presented to vehicles present at different lanes.

Chapter 4

Analysis Modeling

4.1 Activity Diagram

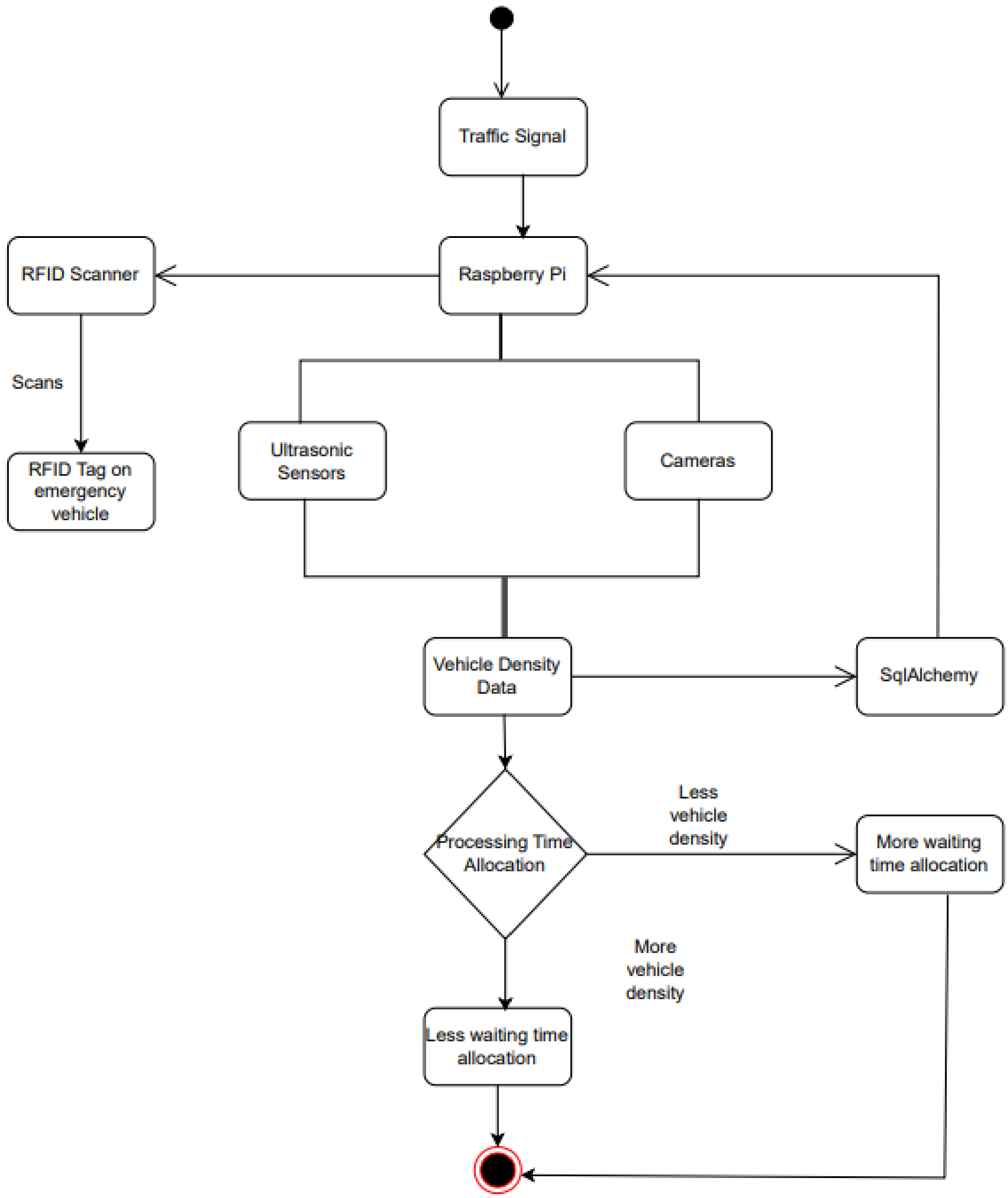


Figure 4.1.1: Activity Diagram

Activity diagram (Figure 4.1) describes the entire flow of the traffic management system. The traffic signal comprises a Raspberry Pi which will have a camera module and will be connected to the ultrasonic sensors. The ultrasonic sensors and the camera module will calculate the vehicle density using image processing via ML algorithms. The calculated data will be constantly stored in SqlAlchemy a database used to store the data to provide static data in case of failure. Now based on the density the ML algorithm will help in estimating the total signal time allocation required for individual lanes. More the vehicle density, the less the allocated time. The Raspberry Pi also consists of an RFID scanner used to track and scan the RFID tags attached to the emergency vehicles.

4.2 Functional Modeling

4.2.1 Data Flow Level 0

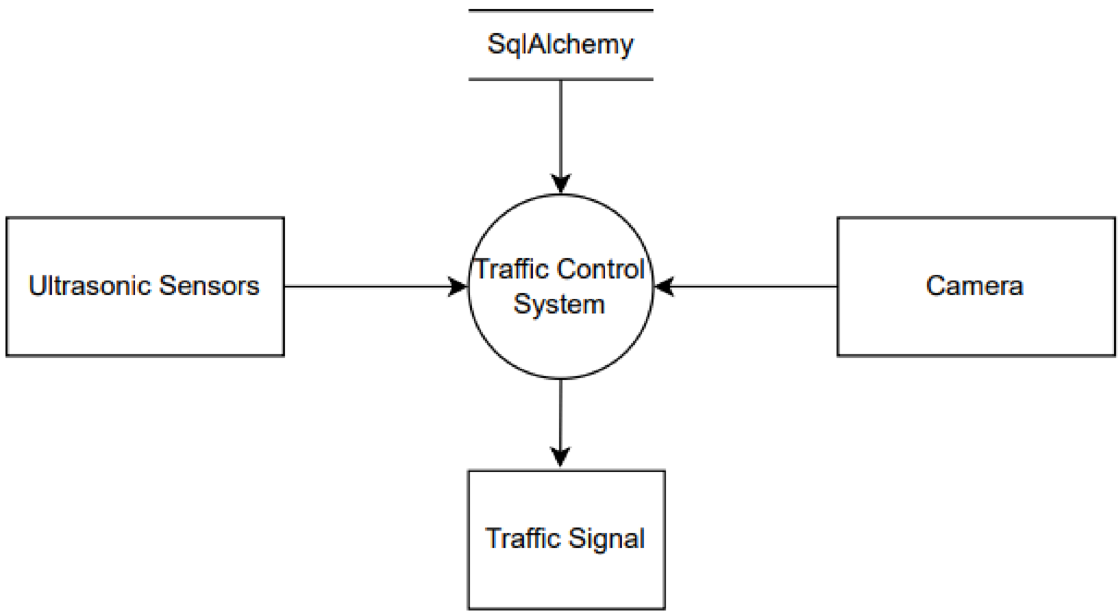


Figure 4.2.1: DFD Level 0

The above diagram is a data flow diagram of level 0. It shows a broad overview of the elements that engage with the system. The camera and ultrasonic sensor accept the input that is the vehicle density data based on object detection. In case of failure of these devices or system failure the inputs will be taken from SqlAlchemy. It then sends the data to the raspberry pi attached to the traffic signal. The traffic signal shows the time according to dynamic allocation based on the amount of traffic in each lane.

4.2.2 Data Flow Level 1

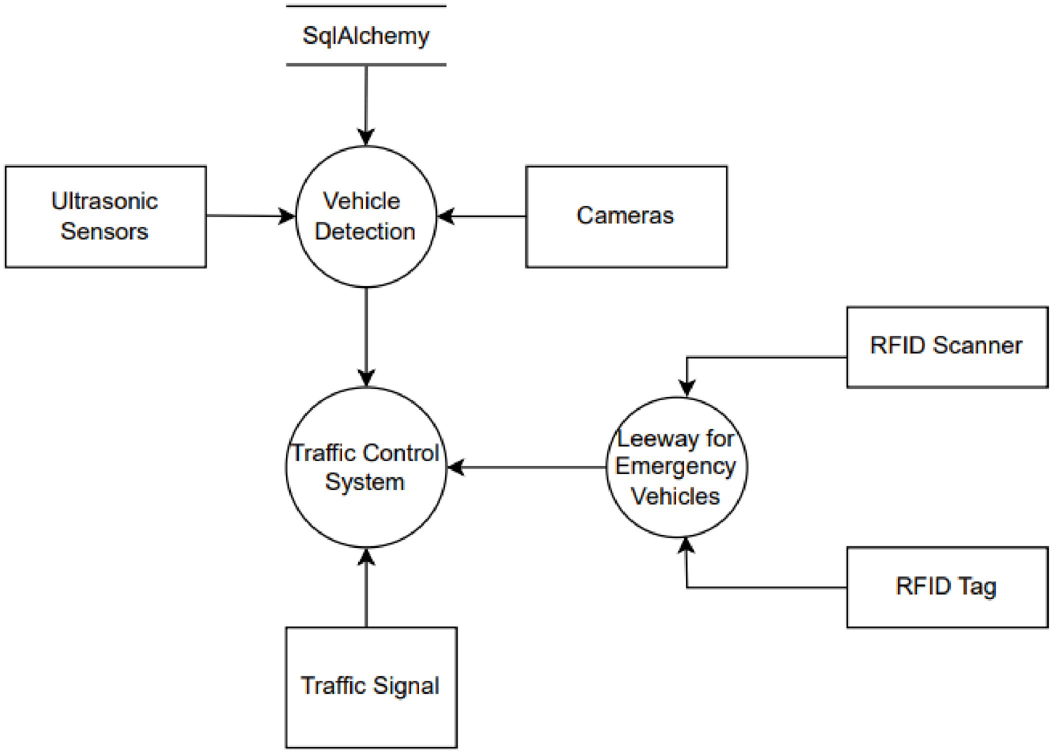


Figure 4.2.2: DFD Level 1

The above diagram is a data flow diagram of level 1. It goes into greater detail about the processes involved and the correlation between the data entities and the processes. The camera and ultrasonic sensor accept the input that is the vehicle density data based on object detection. It then sends the data to the raspberry pi attached to the traffic signal. The traffic signal receives a static data from the database, SqlAlchemy. The traffic signal is connected to an RFID Scanner. The emergency vehicles with RFID tags constantly send data to the RFID Scanner.

Chapter 5
Design
5.1 Architectural design

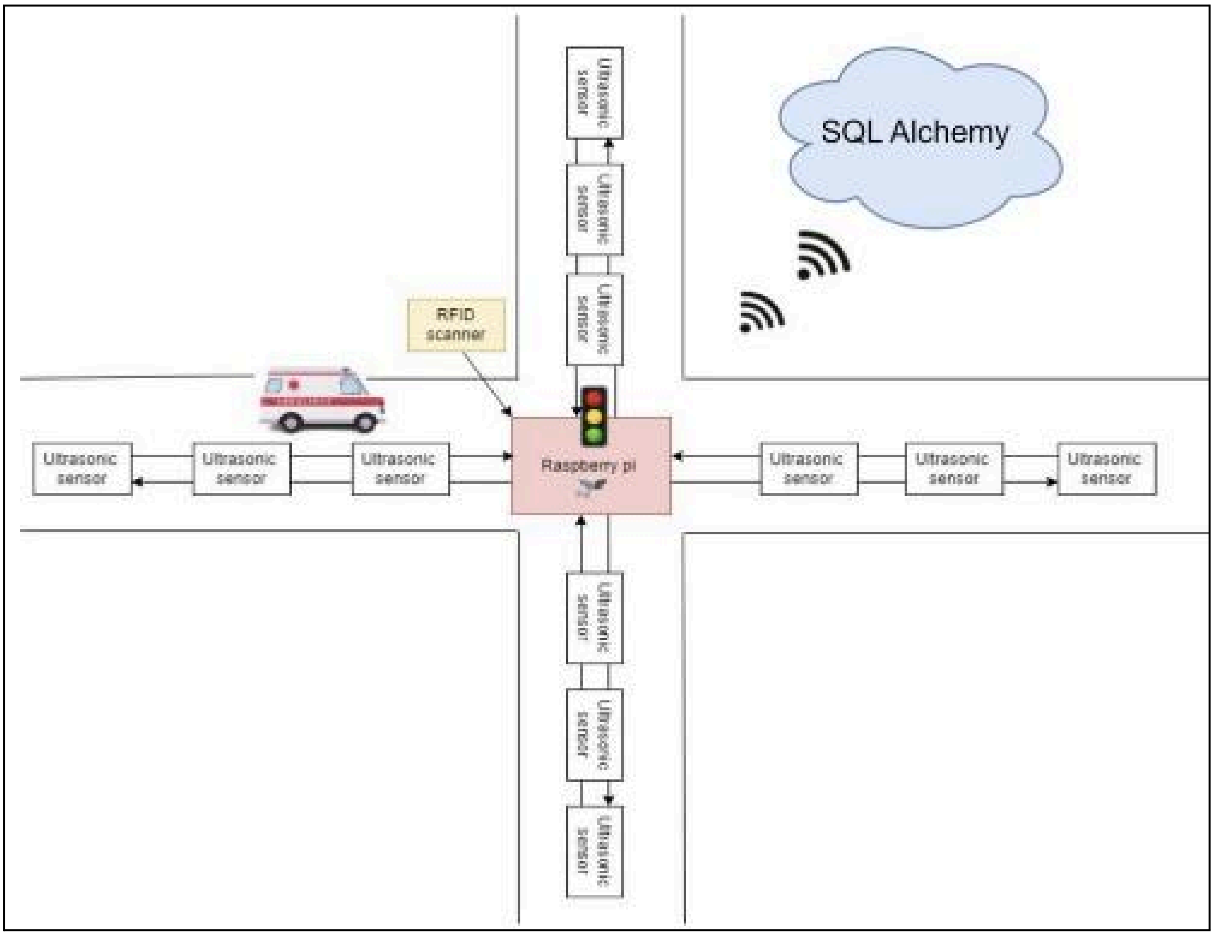


Figure 5.1.1: Architectural design

Chapter 6

Implementation

6.1 High Level Pseudo Code

1. Initialize the system at each junction with the following components:
   - Raspberry Pi for signal control
   - Cameras for image capture
   - Ultrasonic sensors for traffic density measurement

2. Set up the IoT platform (e.g., SqlAlchemy) for data storage and access.

3. Begin the real-time traffic management loop:
   3.1. Capture traffic images using cameras at regular intervals.
   3.2. Process the captured images to identify the number of vehicles and their positions on the road.
   3.3. Use ultrasonic sensors to measure traffic density.
   3.4. Calculate a weighted traffic density score based on image analysis and sensor data.
   3.5. Compute using Raspberry Pi the optimal signal durations based on the traffic density score.
   3.6. Allocate the traffic signal duration based on the processed result by Raspberry Pi
   3.7. Upload the real-time data, including traffic density and signal timings, to the IoT platform.

4. Implement a fail-safe mechanism:
   4.1. Continuously monitor the system's hardware and software for any failures.
   4.2. In the event of hardware failure or extreme discrepancies between camera and sensor data, activate a default signal timing plan to prevent traffic chaos.

5. Periodically check and analyze the data stored on the IoT platform for traffic flow intervals and system performance.

6. Maintain logs and records for system maintenance and further improvements.

7. End the real-time traffic management loop when necessary or as part of regular maintenance.

8. Repeat the loop for each junction equipped with the system.

9. Analyze the data collected for traffic flow improvements and system efficiency.

6.2 Working of Ultrasonic Sensors (HC-SR04)

Ultrasonic sensors operate on the principle of sending and receiving high-frequency sound waves (ultrasonic waves) to measure distances or detect objects. Here's how they work:

a. Sound Wave Generation: The ultrasonic sensor consists of a transducer, which can both transmit and receive sound waves. It generates high-frequency sound waves, typically in the ultrasonic range (above the audible range of human hearing).

b. Sound Wave Transmission: The sensor transmits a short pulse of ultrasonic waves in a specific direction. These waves travel through the air, and their speed is constant at approximately 343 meters per second (in standard atmospheric conditions).

c. Reflection from Object: When the ultrasonic waves encounter an object in their path, they get reflected back towards the sensor. The time it takes for the sound waves to travel to the object and back is measured.

d. Time Measurement: The sensor measures the time it takes for the emitted pulse to bounce back to the receiver. This time is called the "time of flight."

e. Distance Calculation: Using the measured time of flight and the known speed of sound, the sensor calculates the distance to the object. They can be calculated using the following formula:

$$\text{Distance} = (\text{Speed of Sound} \times \text{Time of Flight}) / 2$$

Dividing the result by 2 is necessary because the sound wave travels to the object and then back to the sensor.

f. Output Data: The sensor typically provides distance information in the form of electrical signals, such as voltage or pulse width, that can be processed by a microcontroller or a computer. This information is then used for various purposes, such as object detection or distance measurement.

6.3 Working of Camera Module (RP - 8MP)

The Raspberry Pi camera is a versatile accessory that can capture both still images and video. The 8-megapixel camera module used in Raspberry Pi models, such as the Raspberry Pi 3 and Raspberry Pi 4, operates as follows:

a. Light Sensing: The camera module contains a sensor known as a "Complementary Metal-Oxide-Semiconductor" (CMOS) image sensor. When light strikes the sensor, it converts the incoming light into electrical signals.

b. Bayer Filter: The sensor typically uses a Bayer filter, which is a grid of red, green, and blue color filters placed over the pixels. This filter helps capture color information, as most camera sensors are monochromatic and only sensitive to light intensity. The filter separates incoming light into its RGB components.

c. Pixel Array: The sensor consists of an array of millions of tiny individual light-sensitive elements, known as pixels. Each pixel captures the intensity and color information for the specific region of the image it covers.

d. Analog-to-Digital Conversion (ADC): The analog signals generated by the sensor need to be converted into digital format. The camera module typically has built-in ADC circuitry to perform this conversion.

e. Data Processing: Once the analog signals are digitized, the data undergoes several levels of processing, including demosaicing (to interpolate the RGB colors from the Bayer filter pattern) and compression.

f. Output: The processed image or video data is made available to the Raspberry Pi's central processing unit (CPU) or other connected devices through the Raspberry Pi's camera interface.

6.4 Working of Raspberry Pi 4 (Model B - 8 GB)

The Raspberry Pi, in your traffic management system, serves as the central processing unit and controller, responsible for processing data from the ultrasonic sensors and the camera module, running a trained machine learning (ML) algorithm for traffic classification, and dynamically allocating traffic signal timings based on the detected traffic conditions. Here's an overview of how it works:

a. Data Input:

The Raspberry Pi receives data from two sources:

Ultrasonic Sensors: These sensors provide real-time information about traffic density by measuring distances to vehicles.

Camera Module: The camera captures images of the road or intersection.

a. Image Processing:

The images from the camera module are processed to extract information about the traffic. This may involve various image processing techniques to identify and count vehicles, as well as assess the overall traffic flow.

b. Machine Learning Traffic Classification:

The processed images are input to a trained machine learning algorithm. This algorithm has been trained on a dataset that includes images labeled as "NO," "LOW," or "HIGH" traffic.

The ML algorithm analyzes the features in the images and classifies the current traffic conditions into one of the predefined categories: "NO," "LOW," or "HIGH" traffic.

c. Traffic Signal Allocation:

Based on the classification result from the ML algorithm, the Raspberry Pi dynamically allocates traffic signal timings.

For "NO" traffic, the signal durations for all directions may be reduced to allow for quicker flow.

For "LOW" traffic, signal durations can be adjusted to balance traffic flow and minimize delays.

For "HIGH" traffic, the signal timings can be optimized to reduce congestion and ensure smooth traffic movement.

d. Signal Control:

The Raspberry Pi controls the traffic signals based on the allocated timings. It communicates with the traffic signal lights to adjust the red, green, and yellow light durations for each direction.

e. IoT Data Upload:

The system uploads real-time traffic data, including traffic density measurements from the ultrasonic sensors, images, and the traffic classification result to an IoT platform (e.g., SqlAlchemy).

This data can be accessed for monitoring traffic conditions and for analysis.

f. Fail-Safe Mechanism:

The Raspberry Pi also incorporates a fail-safe mechanism. In the event of hardware failure or extreme discrepancies between the camera and sensor data, a predefined default signal timing plan can be activated to maintain traffic order.

g. Continuous Monitoring:

The Raspberry Pi continuously monitors the system, assesses traffic conditions, and updates signal timings in real-time to adapt to changing traffic patterns.

This approach ensures that traffic signals are dynamically adjusted based on actual traffic conditions, optimizing traffic flow and reducing congestion. The integration of ultrasonic sensors, image processing, machine learning, and the Raspberry Pi's control capabilities allows for an efficient and responsive traffic management system.

6.5 Experiment and Results for Validation and Verification Results :

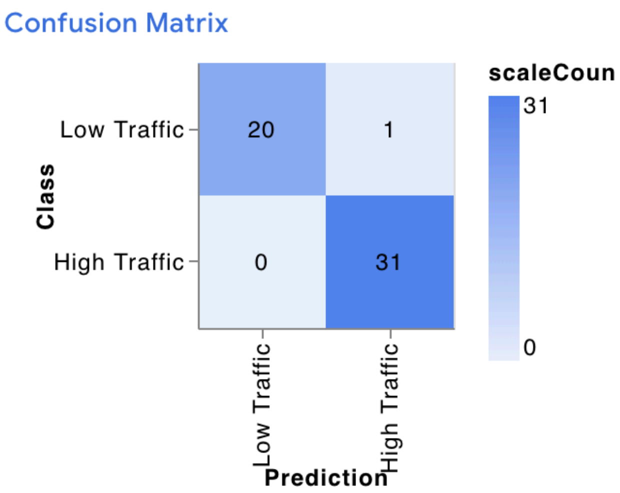
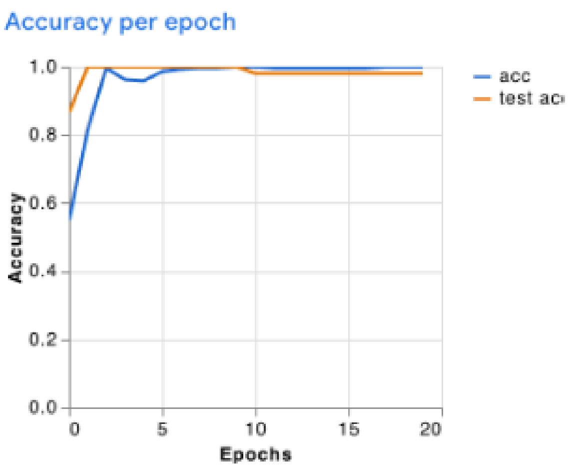


Figure 6.5.1: Performance Measure : Confusion matrix



Accuracy per class

CLASS	ACCURACY	# SAMPLES
Low Traffic	0.95	21
High Traffic	1.00	31

Figure 6.5.2: Performance Measure : Accuracy

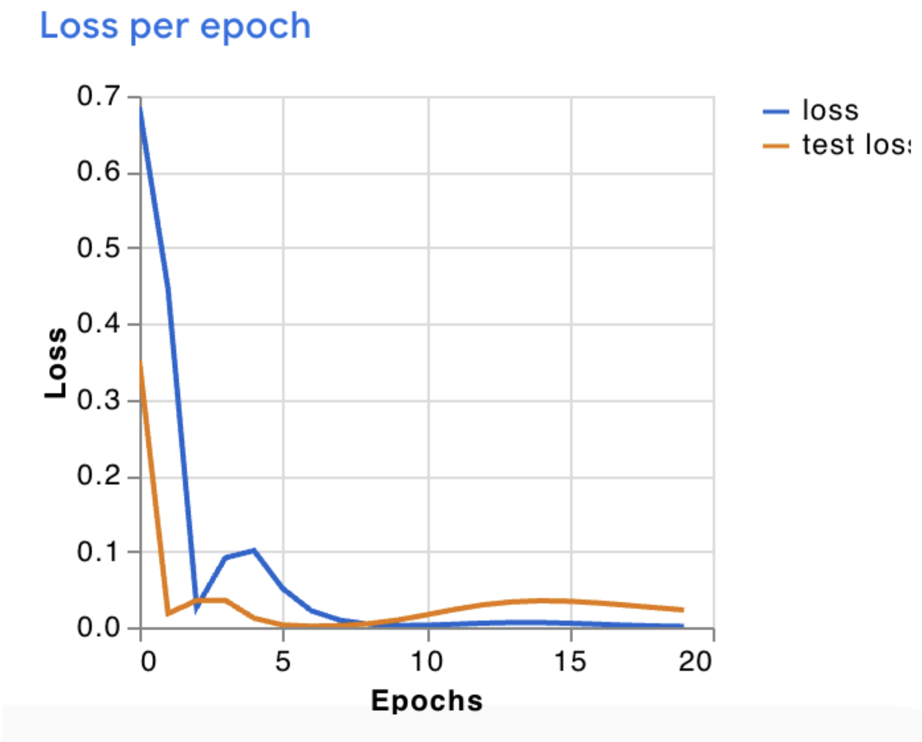


Figure 6.5.3: Performance Measure : Loss

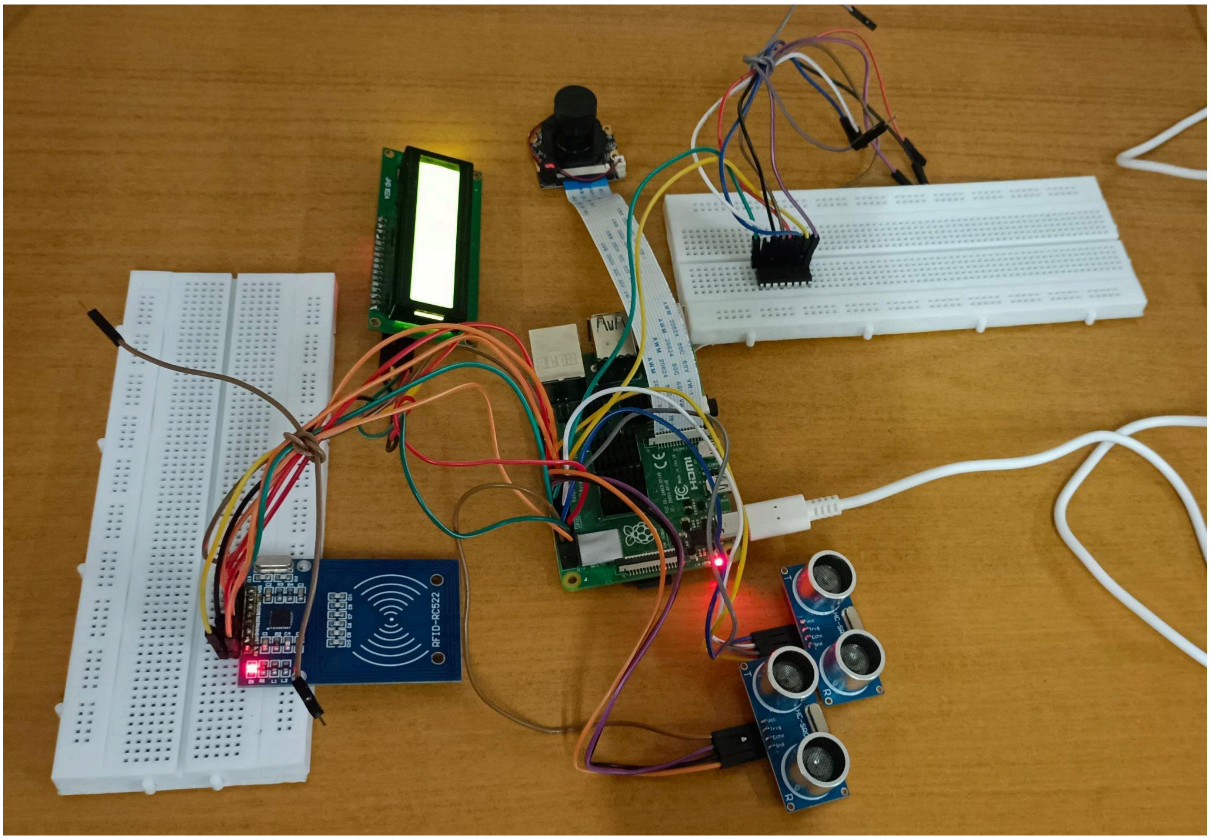


Figure 6.5.4: System design

Figure 6.5.5: Exterior design

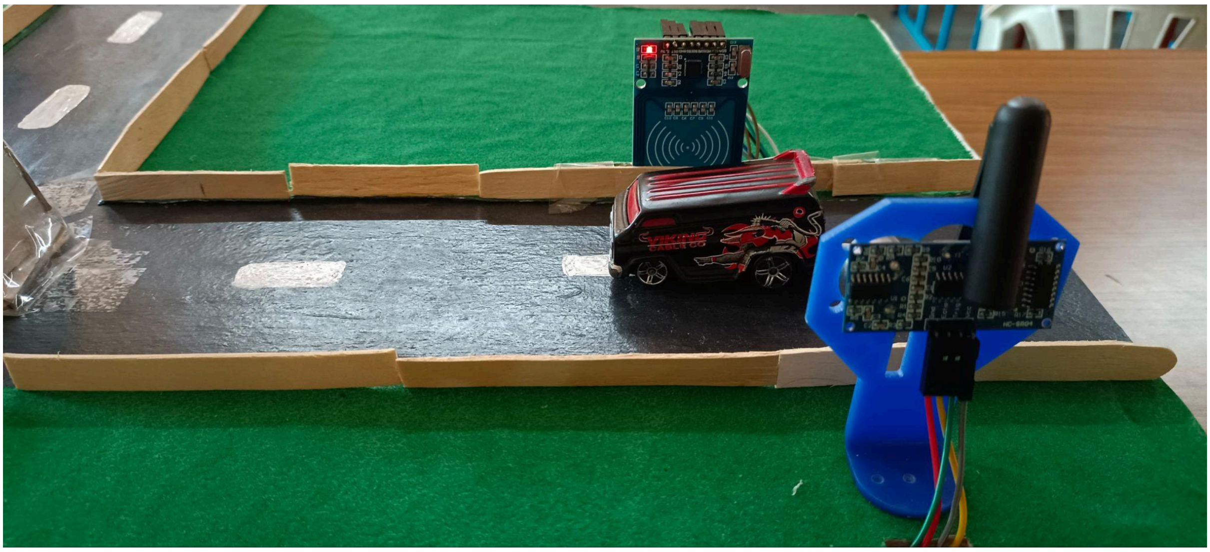


Figure 6.5.6: Working of RFID Scanner

Validations with Test Cases:

```
(myenv) traffix@raspberrypi:~/Downloads $ python image_classification.py
WARNING:tensorflow:No training configuration found in the save file, so
Enter file name without .png:lt6
1/1 [==============================] - 3s 3s/step
[[0.63192993 0.36807    ]]

Your predicted class is  low_traffic
```

```
(myenv) traffix@raspberrypi:~/Downloads $ python image_classification.py
WARNING:tensorflow:No training configuration found in the save file, so th
Enter file name without .png:ht1
1/1 [==============================] - 3s 3s/step
[[0.07681465 0.9231854 ]]

Your predicted class is  high_traffic
```

Figure 6.5.7: Image Classification Model Output

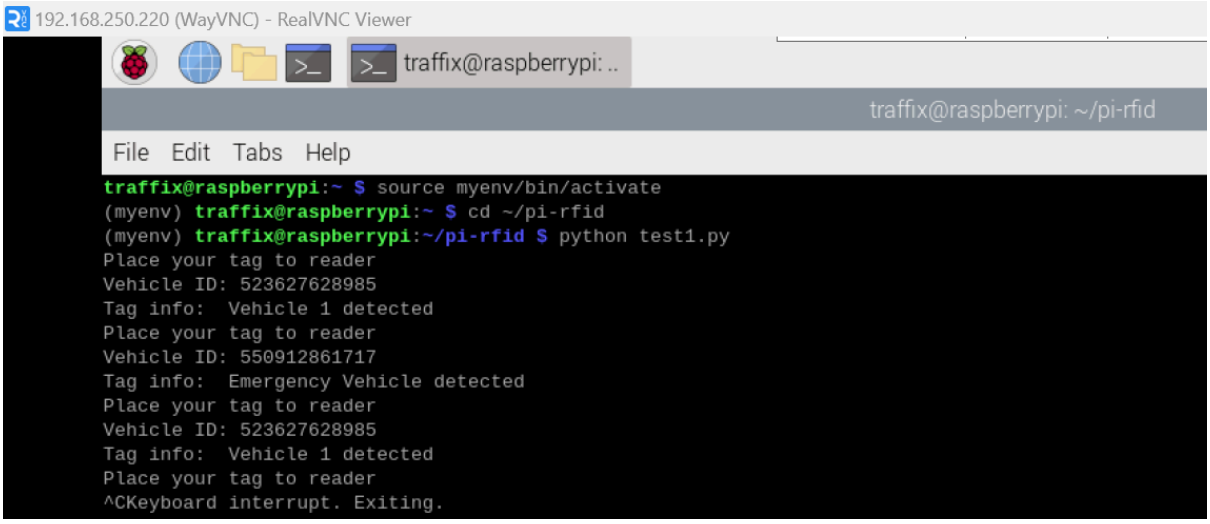


Figure 6.5.8: RFID Output

Test case no.	Image Name	Expected Result	Actual result	Accuracy	Confidence
01	ht-01	High traffic	High traffic	Correct Prediction	81.67%
02	lt-06	Low traffic	Low traffic	Correct Prediction	92.54%
03	ht-05	High Traffic	Low traffic	Incorrect Prediction	57.14%
04	ht-07	High Traffic	High traffic	Correct Prediction	87.63%
05	lt-04	Low Traffic	Low Traffic	Correct Prediction	86.34%
06	lt-03	High Traffic	High Traffic	Correct Prediction	98.44%

Table 6.5.9: Validations with Test Cases

Chapter 7

Conclusion

7.1 Conclusion

The IoT-based dynamic traffic signal time allotment system represents a significant step forward in addressing the challenges of traffic management and congestion within urban environments. By leveraging advanced IoT technologies, data analytics, and real-time responsiveness, the system demonstrates its potential to optimize traffic flow, enhance safety, and improve overall efficiency on roadways. Through the seamless integration of SqlAlchemy and other IoT platforms, the project successfully demonstrated the feasibility and effectiveness of real-time traffic data collection, analysis, and dynamic signal control. The system's adaptive capabilities, coupled with its intelligent traffic monitoring and emergency vehicle prioritization, showcase its potential in reducing travel time, minimizing congestion, and enhancing the overall commuter experience. The project's emphasis on scalability, modularity, and user-friendly interfaces paves the way for future enhancements and customizations, enabling the system to adapt to evolving traffic patterns and urban developments effectively.

7.2 Future Scope

1. Predicting traffic using ML on specific days :

Every year in a specific season people love traveling to certain destinations like for example, people love traveling to Mahabaleshwar or Lonavala for vacations during the rainy season this causes traffic congestion on certain roads. Using machine learning, we can predict such days and traffic patterns to potentially manage it well or minimize congestion.

2. Detecting Illegally parked vehicles :

Vehicles that are parked close to the signal can cause congestion as vehicles do not get adequate space to wait for the signal to open. Such instances can also be picked up using our system and fined accordingly.

References

[1] M. N. Manjunath, S. K. Meghana and S. Rajini, "IoT for Intelligent Traffic Management," 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2019.

[2] D. Ijeri, P. Maidargi and R. Sunagar, "Traffic Control System Using Image Processing," 2020 IEEE Bangalore Humanitarian Technology Conference (B-HTC), Vijiyapur, India, 2020

[3] V. Bali, S. Mathur, V. Sharma and D. Gaur, "Smart Traffic Management System using IoT Enabled Technology," 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCT), Greater Noida, India, 2020

[4] A. M. George, V. I. George and M. A. George, "IOT based Smart Traffic Light Control System," 2018 International Conference on Control, Power, Communication and Computing Technologies (ICCPCT), Kannur, India, 2018

[5] A. Sachenko, O. Osolinskyi, P. Bykovyy, M. Dobrowolski and V. Kochan, "Development of the Flexible Traffic Control System Using the LabView and ThingSpeak," 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Kyiv, Ukraine, 2020

[6] A. Albiol, L. Sanchis, A. Albiol and J. M. Mossi, "Detection of Parked Vehicles Using Spatiotemporal Maps," in IEEE Transactions on Intelligent Transportation Systems, vol. 12, no. 4, pp. 1277-1291, Dec. 2011, doi: 10.1109/TITS.2011.2156791

[7] Umar Farooq, Hafiz Muhammad Atiq, Muhammad Usman Asad, Asim Iqbal, Zeeshan Azmat "Design and Development of an Image processing based adaptive traffic control system with GSM interface”, 978-0-7695-3944-7/10 IEEE DOI 10.1109/ ICMV.2009.65.

[8] Pranav Maheshwari, Deepanshu Suneja, Praneet Singh, Yogeshwar Mutneja, “Smart Traffic Optimization Using Image Processing”, 2015 IEEE 3rd International Conference on MOOCs, Innovation and Technology in Education (MITE). IEEE, 2015

Publications

[1] Reuel Amin, Alarik Correa, Renoy Dsouza, Elvina Fernandes and Prachiti Pimple, "Traffix : Efficient Traffic Control using IoT," 2024 International Journal Of Scientific Research In Engineering And Management (IJSREM), 2024

Copyright

[1] Project title "Traffix : Efficient Traffic Control using IoT", has been submitted for copyright protection. Diary Number: 14114/2024-CO/L