A Preliminary Report on

"Artificially Intelligent Traffic Management System"

SUBMITTED TO THE SAVITRIBAI PHULE UNIVERSITY, PUNE
IN THE PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)
(Academic Year: 2021-22)

SUBMITTED BY

Mr. Saquib Akhtar Aneesur Rahman	**(Exam Seat No. B150474298)**
Mr. Giwil Gidwani	**(Exam Seat No. B150474245)**
Mr. Vrushabh Dattatray Nikam	**(Exam Seat No. B150474278)**
Mr. Rutuja Rajesh Shinde	**(Exam Seat No. B150474304)**

Under the guidance of

Mr. Ravindra Aher

DEPARTMENT OF COMPUTER ENGINEERING

THE MET LEAGUE OF COLLEGES
MET Bhujbal Knowledge City
AS SHARP AS YOU CAN GET

MET's Institute of Engineering,
Adgaon, Nashik-422003
SAVITRIBAI PHULE UNIVERSITY, PUNE

May, 2022

THE MET LEAGUE OF COLLEGES

MET Bhujbal Knowledge City

AS SHARP AS YOU CAN GET

Certificate

This is to Certify that the project report entitles

"Artificially Intelligent Traffic Management System"

Mr. Saquib Akhtar Aneesur Rahman (Exam Seat No. B150474298)
Mr. Giwil Gidwani (Exam Seat No. B150474245)
Mr. Vrushabh Dattatray Nikam (Exam Seat No. B150474278)
Mr. Rutuja Rajesh Shinde (Exam Seat No. B150474304)

are bonafide students of this institute and the work has been carried out by them under the guidance of Mr. Ravindra Aher and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University for the award of the degree of Bachelor of Engineering (Computer Engineering).

Project Guide	H.O.D	Principal
(Mr. Ravindra Aher)	(Dr. M. U. Kharat)	(Dr. V. P. Wani)

Date: / /

Acknowledgements

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individual and organizations. We would like to extend our sincere thanks to all of them. It gives us proud privilege to complete the project on **"Artificially Intelligent Traffic Management System"**. We are highly indebted to our internal guide **Mr. Ravindra Aher** for his guidance and constant supervision as well as for providing necessary information regarding the project and also for his support in completing the project.

We are also extremely grateful to our respected H.O.D. (Computer Department) **Dr. M. U. Kharat** and **Prof. Dr. Priti Metange** (Project Co-ordinator) for providing all facilities and every help for smooth progress of project work.

Mr. Saquib Akhtar Aneesur Rahman
Mr. Giwil Gidwani
Mr. Vrushabh Nikam Dattatray
Ms. Rutuja Rajesh Shinde

Abstract

Congestion of traffic in urban areas and smart cities is one of the major issues with increasing population in metropolitan areas. Traffic jams are not only a cause of delay and inconvenience in day to day life but also a major source of noise and air pollution. Modern approaches to deal with this issue range from complicated software handling dozens of traffic signals throughout an entire city to simpler single-intersection solutions. However these can be costly, difficult to implement and may require a lot of manual monitoring.

This project proposes a traffic management system which uses concepts from artificial intelligence and graph theory to control and optimize traffic flow. Its aim is to optimize traffic flow on a small to medium scale in a manner which adapts to the real time changes in traffic.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter briefly explains the need for an adaptive traffic management system and an overview of the implementation.

1.1 Overview

Traffic congestion is becoming one of the critical issues in cities with increasing population and number of vehicles. They not only cause problems like delays and stress to drivers but also cause secondary problems like increasing fuel consumption, transportation costs and pollution.

The causes of congestion can be divided into two categories, recurring and non recurring congestion. Recurring congestion can be expected to occur at the same time every weekday as a result of high volumes of commuter traffic traveling on roadways that are at or near their carrying capacity. Non-recurring congestion occurs as a result of an unexpected or non-typical event. Some causes of non-recurring congestion include: vehicular crashes, vehicle breakdowns, roadway construction, inclimate weather, and additional traffic resulting from special events. While non-recurring congestion can be unpredictable and difficult to treat, recurring congestion can be reduced by increasing road capacity or with the help of adaptive traffic control systems.

There are several existing standardized solutions for adaptive traffic control such as SCOOT[5], SCAT[6], etc. which have been implemented in many major metropolitan cities. However, most suburban and urban areas use conventional traffic control systems such as manual traffic control or non adaptive automated traffic control. Manual control

consists of an on-site traffic official guiding vehicles. Non adaptive automated traffic control refers to the use of fixed timers in traffic signals. Wide implementation of standardized adaptive traffic control is not possible due to lack of feasibility since it requires manual labor and installation of new sensors. Therefore a more feasible solution which reuses existing infrastructure is required.

This project proposes an Artificially Intelligent Traffic Management System which uses existing CCTV feed and API data if needed to optimize traffic control over small to medium scale road networks. It uses an artificially intelligent agent or model to handle the complexity of day to day traffic in real-time. Furthermore, simulations will be performed to demonstrate and test the model.

1.2 Summary

This chapter discusses the need for an intelligent traffic management system and also discussed a brief overview of the project.

Chapter 2

Literature Survey

This chapter consists of the various studies and research conducted on key concepts which are essential to create and understand the proposed system.

2.1 You Only Look Once: Unified, Real-Time Object Detection, Joseph Redmon, et al. [1]

Object detection comprises locating specific types of objects in an image or video. The output of a typical object detection algorithm consists of bounding box coordinates and a label of the object. YOLO (You Only Look Once) model consists of an extremely fast unified architecture for object detection. It makes use of a single neural network for predicting bounding boxes and class probabilities from a full image in a single evaluation. Hence, making it ideal for object detection in real time applications.

2.2 Traffic Congestion Detection from Camera Images using Deep Convolution Neural Networks, Pranamesh Chakraborty, et al. [2]

Recent improvements in computer vision algorithms have led to closed-circuit television (CCTV) cameras emerging as an important data source for determining the state of traffic congestion. To detect congestion in a traffic CCTV footage YOLO, a

state-of-the-art real-time object detection algorithm is used. In the above mentioned paper, several object detection techniques were tested for congestion detection out of which YOLO showed the most promising results.

2.3 Smart Control of Traffic Light Using Artificial Intelligence, Mihir M. Gandhi, et al. [3]

Traffic signal timing plays an important role in controlling flow and efficiency of traffic. The above system makes use of vehicle count obtained from CCTV footage and uses it to optimize green signal timing for each lane to optimize traffic flow at a single intersection. The aim of the project is to create a similar system and extend the scope of optimization to multiple adjacent intersections.

2.4 Comparison of Current Practical Adaptive Traffic Control Systems, Hongyun Chen, et al. [4]

Existing Adaptive Traffic Control Systems (ATCS) such as SCOOT, SCAT, OPAC, RHODES, etc. are being adapted by major cities in developed and developing countries. They can cover up to hundreds (OPAC) to even thousands (SCOOT, SCAT) of intersections. However, their implementation includes installation of additional sensors and can lead to very heavy costs which isn't feasible for smaller cities. Additionally, these systems do not take into consideration challenges such as power failure, non lane following traffic and mixed traffic which are common in Indian roads.

The aforementioned systems provide key algorithmic insights for developing a more feasible ATCS model. Furthermore, usage of existing inputs such as CCTV needs to be emphasized over installation of new sensors in order to help reduce costs.

2.5 Comparison of PPO and SAC Algorithms Towards Decision Making Strategies for Collision Avoidance Among Multiple Autonomous Vehicles, Abu Jafar Md Muzahid, et al.[7]

Commonly used reinforcement learning algorithms, namely PPO (Proximal Policy Optimization) and SAC (Soft Actor-Critic) were trained, evaluated and compared using Unity3D and ML-Agents.

2.6 A Deep Reinforcement Learning Approach for Traffic Signal Control Optimization, Zhenning Li, et al. [8]

This paper demonstrates optimization of traffic signal control using reinforcement learning. Training a deep neural network using reinforcement learning can help improve adaptability od the signal.

2.7 Summary

This chapter reviews research done on key concepts such as object detection and implementation of artificial intelligence with traffic signals which are essential to the proposed system. Existing solutions, their inner workings as well as their advantages and disadvantages were studied.

Chapter 3

Problem Definition

This chapter discusses the drawbacks of current systems implemented in suburban and urban areas and also defines the need and overall scope of an artificially intelligent traffic management system.

3.1 Need For Artificially Intelligent Traffic Management Systems

Traffic congestion is becoming a critical issue with increasing population and automobiles in cities. The conventional systems which were suitable at the time of their installation may not be suitable in the present time due to the rising number of vehicles. Furthermore, upgrading these systems to the standards used in major metropolitan cities is often not feasible due to several factors such as manual labor and installation of new sensors. Owing to these factors, the majority of intersections make use of either manual control which includes traffic police officials guiding vehicles or non adaptive automatic control, which includes the use of fixed timers. In most scenarios these solutions may not be at par with the unpredictable rate of traffic flow. Hence, an intelligent, adaptive and feasible solution is needed.

3.2 Additional Features

Additionally, the following points must be kept in mind while developing or optimizing an automated traffic controlling system:

- Ensuring signal times are between a maximum and minimum limit to avoid starvation.
- Indian traffic is not lane following and has a high amount of mixed traffic. System must be able to withstand these challenges.
- The system will be able to use the existing sensors and avoid installation of newer sensors in order to maintain feasibility.
- The system will be scalable within budget and it should be able to handle an increasing number of vehicles.
- Additional features such as incident detection, report generation and assistance for emergency or VIP vehicles can be added.

3.3 Summary

This chapter discusses the need of artificially intelligent traffic management systems and the various points which must be kept in mind while developing said system.

Chapter 4

Analysis

This chapter describes the project plan adopted and determines the requirement analysis. The project was implemented on the basis of Agile model and Model View view-model.

4.1 Project Plan

4.1.1 Project Plan for semester I

The following Table 4.1 describes the project plan for semester I. It describes the various activities and accountability of the developers for the respective modules. Following are the major activities carried out in this plan :

- Identifying the functional requirements.
- Designing of the Framework.
- Studying the necessary development tools and technologies.

Phase	Activity	Start Date	End Date	Group Members
1	Selection of Project Topic	06-09-2021	13-09-2021	Team
1	Functional Requirement Specification(FRS)	19-09-2021	03-10-2021	Team
1	Design Prototype	11-10-2021	21-10-2021	Team
1	Graph Theory and Math Model	23-10-2021	06-11-2021	Saquib, Giwil
1	UML Diagram Prototype	23-10-2021	03-11-2021	Vrushabh, Rutuja
1	Project Problem Statement using NP Complete	08-11-2021	19-11-2021	Saquib, Giwil
1	UML Diagram in StarUML	20-11-2021	22-11-2021	Team
1	Presentation	05-11-2021	08-11-2021	Team
1	Software Requirement Specification	6-12-2021	10-12-2021	Team

Table 4.1: Planner and Progress Report I for AITMS

4.1.2 Summary for semester I

In this chapter we described the implementation details of the project plan for Semester I. We also studied the functional requirement specification, design prototype, and project problem statement using NP completeness of AITMS.

4.1.3 Project Plan for semester II

The following Table 4.2 describes the project plan for semester II. It describes the various activities and accountability of the developers for the respective modules. Following are the major activities carried out in this plan :

- Define Programming Standards.
- Development of project in 3 Milestones.
- Formal Technical Review and Testing.

Phase	**Activity**	**Start Date**	**End Date**	**Group Members**
2	Defining Programming Standards	01-01-2022	09-01-2022	Team
2	Development of Milestone No.1	10-01-2022	28-02-2022	Team
2	Development of Milestone No.2	7-03-2022	03-04-2022	Team
2	Development of Milestone No.3	04-04-2022	30-04-2022	Team
2	Formal Technical Review	04-05-2022	06-05-2022	Team
2	Testing and Bug Fixing	09-05-2022	12-05-2022	Team

Table 4.2: Planner and Progress Report II for AITMS

4.1.4 Summary for semester II

In this chapter we described the implementation details of the project plan for Semester I. We also studied the necessary functions and the desirable functions of AITMS.

4.2 Requirement Analysis

4.2.1 Necessary Functions

- Deliver a reusable piece of code.
- Producing optimized traffic signal timing as output
- Deployment of simulation built onto the unity.

4.2.2 Desirable Functions

- Assistance to emergency and VIP vehicles.
- Traffic incident detection.
- Real-time Traffic Congestion Detection.
- Real-time Statistics.

4.3 Summary

This chapter describes the implementation details of the project plan for Semester I. The necessary functions and the desirable functions of Artificially Intelligent Traffic Management System were also studied.

Chapter 5

Design

This chapter describes the Software Requirement Specification (SRS) to be implemented for Artificially Intelligent Traffic Management System. It also explains the architecture of the system and external interface requirements.

5.1 Software Requirement Specifications

The Software Requirement Specification describes the scope of the project, operating environment, user characteristics, design and constraints. It also elaborates the system architecture of the Artificially Intelligent Traffic Management System.

5.1.1 Project Scope

The main purpose of developing the Artificially Intelligent Traffic Management System is for the welfare of the public and to reduce the pollution caused by traffic congestion. The issue of transportation is one of the most problematic issues in the country, and in the world at large. In an age where a person has a private car, and especially in countries where there is a failing public transport system, traffic jams and the countless accidents that accompany road congestion harm the economic, political and social aspects in both the public and private sectors. the transportation systems used by transportation agencies around the world are not adapted to modern transportation - these are systems designed decades ago, when the amount of cars on the roads was much smaller than today, and the transportation nodes were relatively simple. today,

innovative technologies for computerized transportation management are in use, but these turned out to be extremely expensive, and due to budget shortages and even many cuts in the transportation budget, transportation agencies are often forced to give up these systems and stick with the old systems. Countries such as Australia, Singapore, and a number of U.S. states have adopted transportation management technologies on a massive scale (for example, GLIDE, SCATS[6] - these are huge economic investments, these are proving themselves as highly prudent and effective investments. The purpose of the software is to enable the optimization of traffic signals through implementation of machine learning on data collected at traffic signals, making them more adaptive to real time changes in traffic flow. The software is made to be intuitive and accessible as much as possible, while maintaining accuracy, maximum detail, level of performance And high efficiency.
The scope of Artificially Intelligent Traffic Management System can be considered as a collection of reusable piece of code, 3D roadway components, trained models and include files that can be used by the developer for developing more advanced systems and also considering the following points :

- Warehousing of parametric data and reports to analyze periodic patterns and improve optimization using it (data science)
- Making AITMS compatible with each other so that neighboring solutions can work together providing another layer of optimization with added scalability
- Providing a platform for simulating roadmaps and training smart signals; allowing the user maximum flexibility in design of maps and smart signal models

5.1.2 Operating Environment

The proposed system will require an operating environment with appropriate version of Windows or Linux OS, Python, Unity3D and sufficient computation power through GPUs.

5.1.3 Design and Implementation Constraints

The key restriction here will be to verify the validity of the report, which is not always feasible.Security threats may be involved.

- **Memory:** Minimum 16GB RAM
- **CPU:** Intel Core i5 - 8300H or equivalent\higher.
- **GPU:** Nvidia GTX 1050ti or equivalent\higher.
- **LAN for CCTV:** CCTV feed from intersections to the server room.
- **Operating System:** This application works on Windows and Linux.

5.1.4 Assumptions

AITMS functions based on the following assumptions.

- Inputs such as vehicle count, phase intervals and current phase are obtainable from the junction in real time.
- It is possible to interface traffic signals with AITMS in a way which allows AITMS to set phase intervals and traffic signal state for each lane.
- The Admin has a computer with graphic card, gets to monitor intersections and rest of the system is automated.

5.1.5 Dependencies

AITMS requires the following software dependencies to be fulfilled for proper functioning.

- **Python v3.7.13**
  Python is a high-level, interpreted, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Python is dynamically-typed and garbage-collected.
- **Unity3D v2020.3.32f**
  Unity is a game development platform. Unity is used to build high-quality 3D and 2D games and simulations, deploy them across mobile, desktop, VR/AR, consoles or the Web, and connect with loyal and enthusiastic players and customers.

- **Unity ML-Agents v1.0.8**

  The Unity Machine Learning Agents Toolkit (ML-Agents) is an open-source project that enables games and simulations to serve as environments for training intelligent agents using deep reinforcement learning and imitation learning.

- **PyTorch v1.8.1**

  PyTorch is an open source machine learning framework based on the Torch library, used for applications such as computer vision and natural language processing, primarily developed by Facebook's AI Research lab. It is free and open-source software released under the Modified BSD license.

5.2 System Architecture

The overall architecture can be viewed in the figure 5.1 in the form of a 3 layer architecture consisting of physical intersection, intersection node and the optimizer.

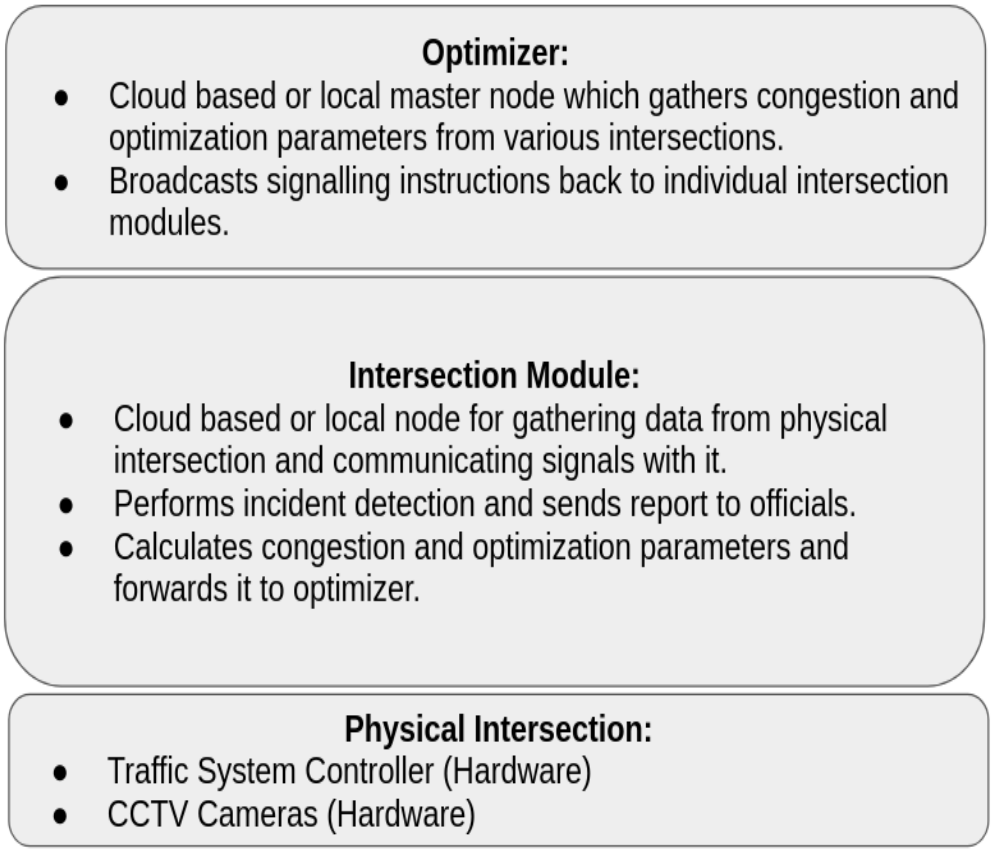

Figure 5.1: Layered form of System Architecture

The lowermost layer i.e. the physical intersection consists of the physical traffic signal controllers and CCTV cameras. This layer contains the pre-existing infrastructure used for controlling traffic. The traffic signal controller must provide an interface to obtain and control its state. Furthermore, real-time CCTV footage must be made available to the model.

The intermediate layer consists of intersection nodes (software). Each intersection node represents a single traffic intersection. Intersection nodes serve the following primary functions:

- Collecting CCTV footage and other data from physical intersection layer and APIs (if required by optimizer).
- Obtaining optimizer parameters from the collected raw data. For example, obtaining vehicle count from CCTV footage using YOLO. The intersection node then forwards this data to the optimizer.
- Intersection nodes also receive timing signals as outputs from the optimizer, their task is to then interface these signals onto the controller.

Aside from the above functions, the intersection nodes can also be used to perform extra tasks such as incident detection and sending reports on traffic incidents and congestion to traffic officials. It must be noted that since intersection nodes are services in execution, they can be executed over cloud or locally depending on feasibility. Depending on availability of computation power and hardware setup, a single computing device can hold multiple intersection nodes as well.

The uppermost layer consists of the optimizer. The optimizer is the master node which receives the required data from all the intersection nodes and executes the optimization algorithm. The optimization algorithm gives the signal timing details as output in terms of green signal time which is broadcasted back to the respective intersection nodes.

Several optimization algorithms were selected which can be applied depending on their performances on a simulation consisting of a specific map. It should also be noted that different optimization algorithms may be suited for different maps.

The following figure 5.2 illustrates the system architecture for 3 intersections.

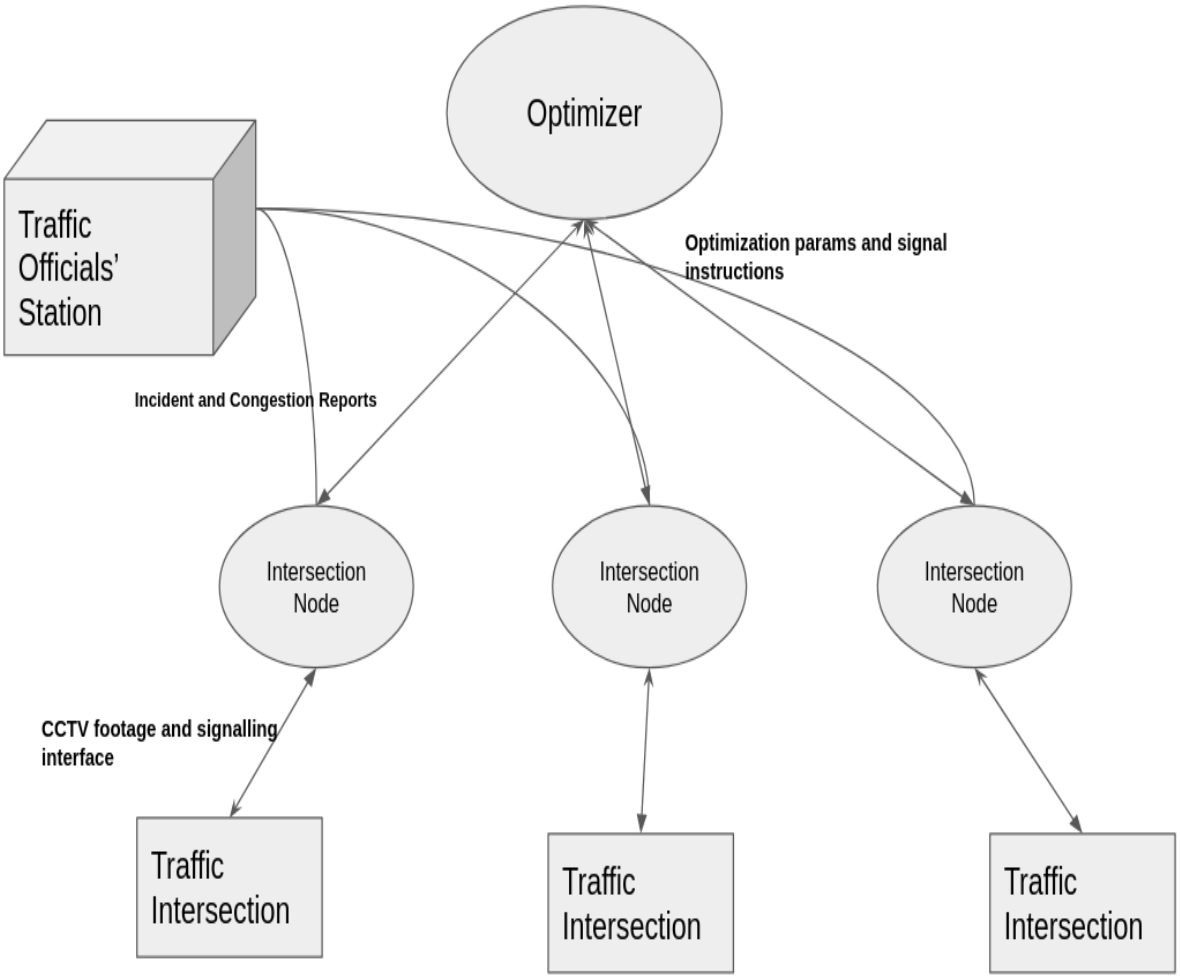

Figure 5.2: System Architecture Illustration

5.3 Software System Attribute

- **Reliability:** The traffic management system should be reliable with necessary fallbacks in case of technical failuers.
- **Maintainability:** The Artificially Intelligent Traffic Management System shall be well documented and easy for developers to improve on

5.4 Data Flow Diagram

The Data Flow Diagram explains the flow of information in the project, i.e. it indicates from where data or information is received (input) and to where it is sent (output). The Data Flow Diagrams for the project are given idn figure 5.3 and figure 5.4:

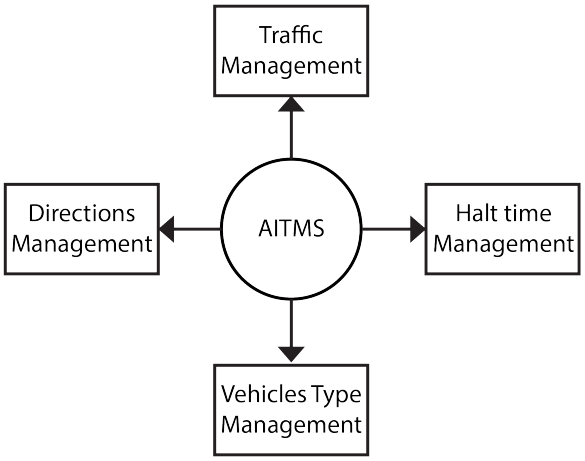

Figure 5.3: Level 0 Data Flow Diagram

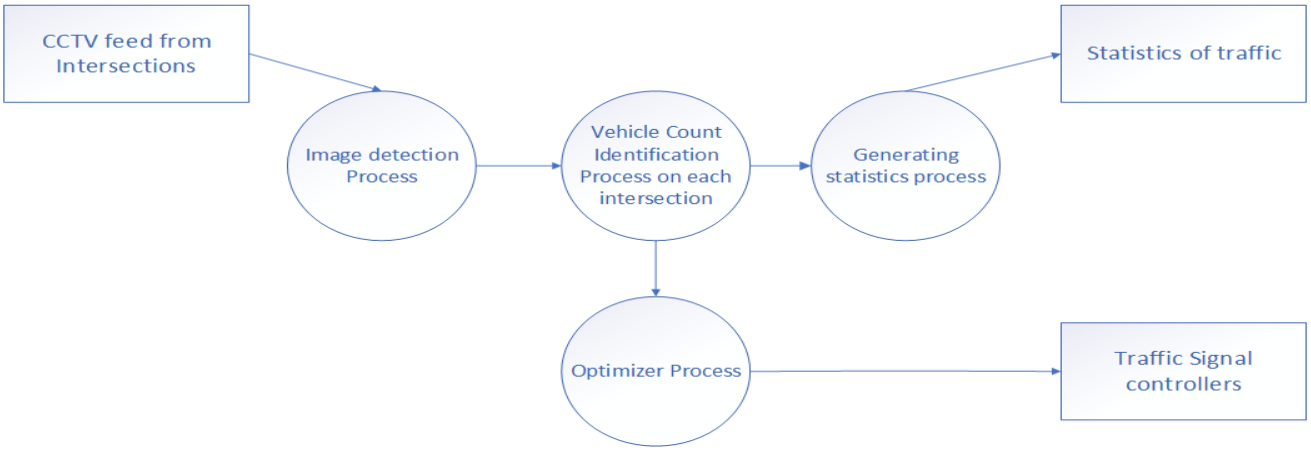

Figure 5.4: Level 1 Data Flow Diagram

5.5 Summary

This chapter discusses the system architecture, operating environment and the software attributes which describe the scope of the project.

Chapter 6

Modeling

This chapter includes the various modeling techniques which describes the various users of the Artificially Intelligent Traffic Management System. It also describes the functionality of the different features of the Artificially Intelligent Traffic Management System.

6.1 Use Case Diagram

A use case diagram is a type of behavioral diagram defined by the UML created from a use case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals represented as use case and any dependencies between those use cases.

Four modeling elements make up the use case diagram; these are:

- **Actors:** Actors refer to a type of users, users are people who use the system. In this case traffic officers are the users of the framework and application
- **Use cases:** A use case defines behavioral features of a system. Each use case is named using a verb phrase that express a goal of the system. The name may appear inside or outside the ellipse.
- **Associations:** An association is a relationship between an actor and a use case. The relationship is represented by a line between an actor and a use case.

- **The include relationship:** It is analogous to a call between objects. One use case requires some type of behavior which is fully defined in another use case.
- **The extend relationship:** It is intended for adding parts to existing use cases as well as for modeling optional system services

The usecase diagram for proposed system is given in the figure 6.1.

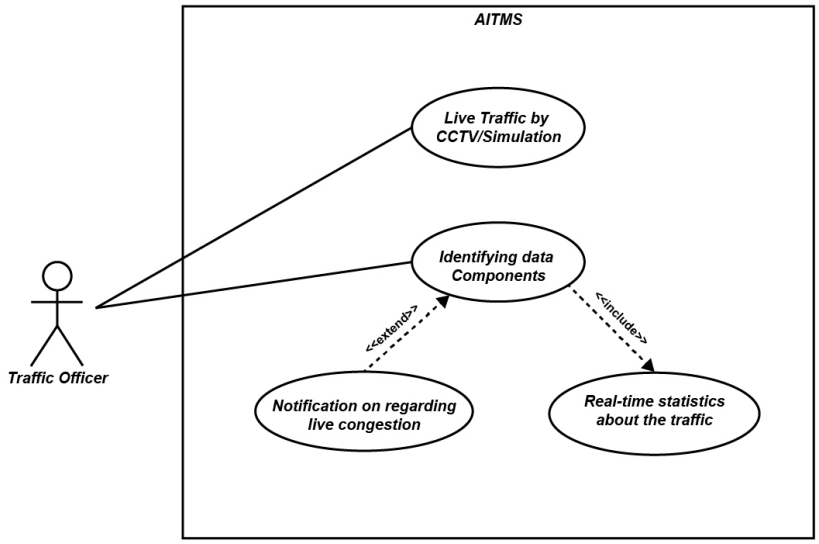

Figure 6.1: Use Case Diagram

6.2 Activity Diagram

Use cases show what your system should do. Activity diagrams allow you to specify how your system will accomplish its goals. Activity diagrams show high-level actions chained together to represent a process occurring in your system. An activity diagram is essentially a flowchart, showing flow of control from activity to activity. Unlike a traditional flowchart, an activity diagram shows concurrency as well as branches of control. Activity diagrams focus on the dynamic flow of a system.
The activity diagram for proposed system is given in the figure 6.2.

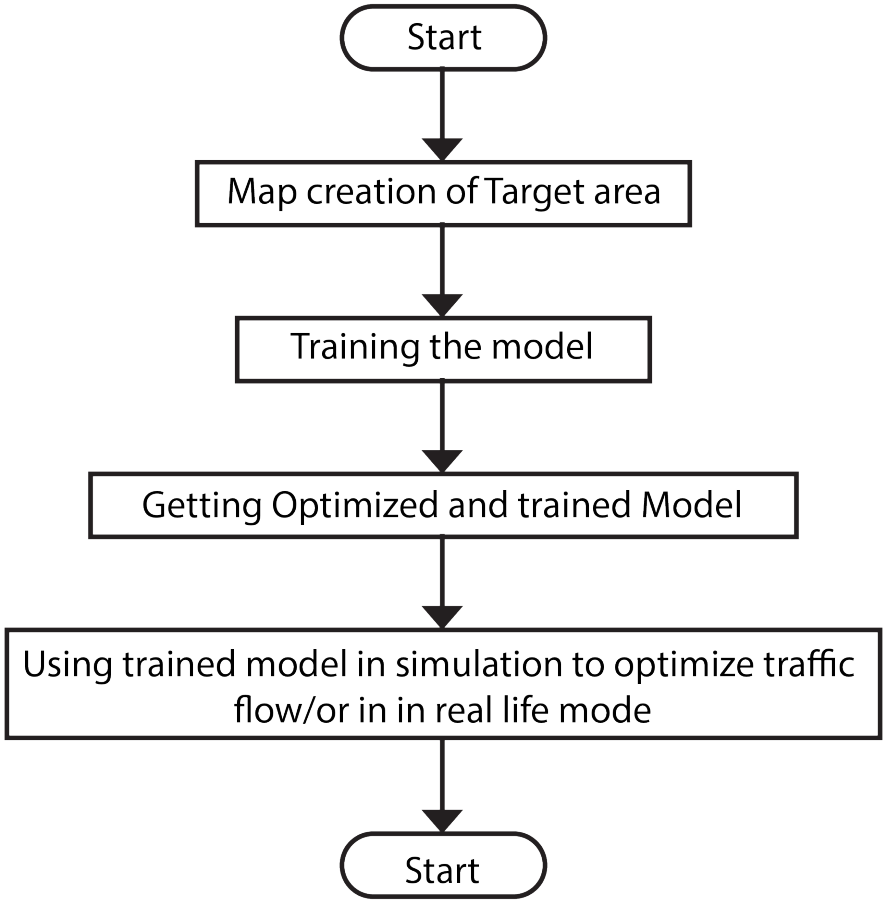

Figure 6.2: Activity Diagram

6.3 Component Diagram

Component diagram are one of the two kinds of diagrams found in modeling the physical aspects of object oriented systems. A component diagram shows organization and dependencies among set of components. Component diagram can be seen to model the static implementation view of a system. This involves modeling the physical things that resides on a node, such as executables, libraries, tables, files and documents.

Component diagram shows a set of components and their relationships. Graphically a component diagram is a collection of vertices and arcs. Component diagrams commonly contain,

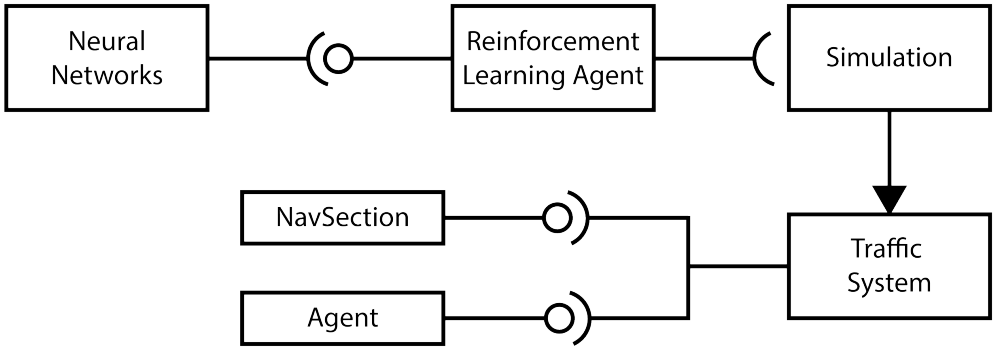

Figure 6.3: Component Diagram

6.4 Summary

Thus the various modeling techniques used for the design of Artificially Intelligent Traffic Management System were seen in this chapter.

Chapter 7

Implementation and Results

This chapter consists of the various implementation details and snapshots of the Artificially Intelligent Traffic Management System.

7.1 Implementation Details

- **Simulation** The Simulation is one of the most important features provided by AITMS. Simulation provides creation of maps of traffic system, it also helps in generating real-time traffic and visualizing traffic densities. It is implemented in Unity3D. Simulation is further divide into many smaller modules below.
  - **NavMesh Module:** The NavMesh represents the area where the center of the agent can move.

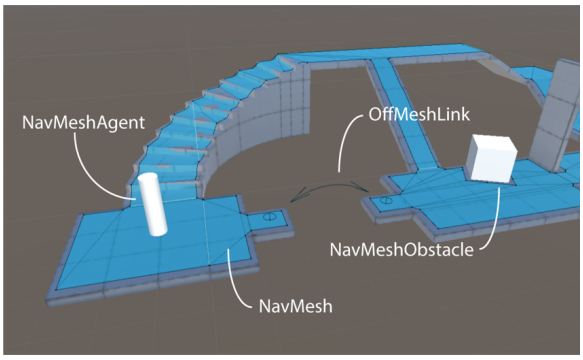

Figure 7.1: NavMesh Components

Conceptually, it doesn't matter whether you regard the agent as a point on a shrunken NavMesh or a circle on a full-size NavMesh since the two are

equivalent.NavMesh module help in baking the path for agents to run on the guided path which makes it easier to simulate real-time traffic.

- **Agent Module:** In this module, Agents(Different vehicles such as cars, buses, trucks etc) prefabs are available. These agents prefabs are used to simulate vehicles in the real-time traffic system.
- **NavSection Module:** In this module, NavSection prefabs are available which helps in designing realistic maps. It includes many predefined NavSection prefabs for single lane, double lane, bridges, highway roads etc.
- **Junction Observables Module:** This module observes the current state and environment of the juntions and passes it to RL Agent. This module is used only when the model is training. It outputs current phase, phase count, phase timing to RL Agent.

- **Reinforcement Learning**

  It is an area of machine learning concerned with how intelligent agents ought to take actions in an environment in order to maximize the notion of cumulative reward.

  The model is trained using MLAgents toolkit in Unity. Training is set up by adding the JunctionRLAgent script to the Junction object which is to be trained. JunctionRLAgent consists of the reward mechanism which penalizes (i.e. negative reward) the model based on halt duration of vehicles that pass through the junction and the number of vehicles remaining in a lane after its phase.

  The aim of training would be to minimize these parameters, hence model must be able to reduce the overall waiting time of vehicles and at the same time provide optimal green time for all waiting vehicles of the current phase to pass through. The input of the model is obtained from the Junction Observables Module explained above.

7.2 Steps for Training a Smart Traffic Signal

Following steps are to be followed to train a smart signal from scratch using AITMS.

- Create a new scene in Unity3D; Create a road map consisting junctions by simply dragging and dropping NavSection prefabs. Finally click bake to complete map creation.
- Tune generalized parameters (speed limits, connections, etc.) of placed NavSection components and vehicle prefabs as desired.
- Add JunctionRLAgent script to junctions which are to be trained. Optionally, pass GlobalObservables if inference from other junctions is required. If a custom reward mechanism is desired, it can be implemented in the Junction script.
- Edit training configuration file as desired.
- Start an instance of MLAgents using the configuration file and hit play in Unity.
- Let model train for atleast a couple thousand steps, this step takes the longest time depending on size of map, complexity of reward mechanism and training configuration. Training can be visualized as it happens using tensorboard.
- Attach trained model (".onnx" file) to Junction in scene and evaluate by enabling log option in AITMS object.

7.3 Training Configuration

MLAgents models are trained based on a configuration file having a ".yaml" extension. Properties for each model such as trainer type, number of layers, etc. are defined in this file. It allows tuning hyperparameters and environment settings. For more details check out the MLAgents documentation page.

7.4 Reward Mechanism

Reward mechanism plays a key role in training reinforcement learning models. The aim of reinforcement learning is to maximize rewards obtained through each decision made. The default reward function of AITMS is given as follows

$$\textbf{Reward} = -(\ \textbf{AvgHaltTime} \times \textbf{LaneCount})$$

where, AvgHaltTime is the average of waiting time of all the vehicles that have passed through the junction in current phase and LaneCount is the amount of vehicles (value is scaled using maxmin scaling) remaining after end of current phase. Reward is calculated at the end of each phase.

Since reward is negative, the model tries to minimize the magnitude during training. Hence, through training, the model should be able to reduce average waiting time of the vehicle as well as set optimal phase time such that maximum waiting vehicles pass through during every phase.

7.5 Model Architectures

AITMS models can be trained in the following possible structures.

- The same model is attached to multiple junctions. Hence junctions share a generalized model.
- Different model is attached to each junction. Hence each model is specialized to optimize its own junction.
- Hybrid, i.e. a combination of the above two structures.

7.6 Results

The snapshots below are taken on the desktop device itself having Windows 11 or Linux based operating system, screen of 17 inches with a resolution of 1900 x 1080.

Following are the snapshots of AITMS:

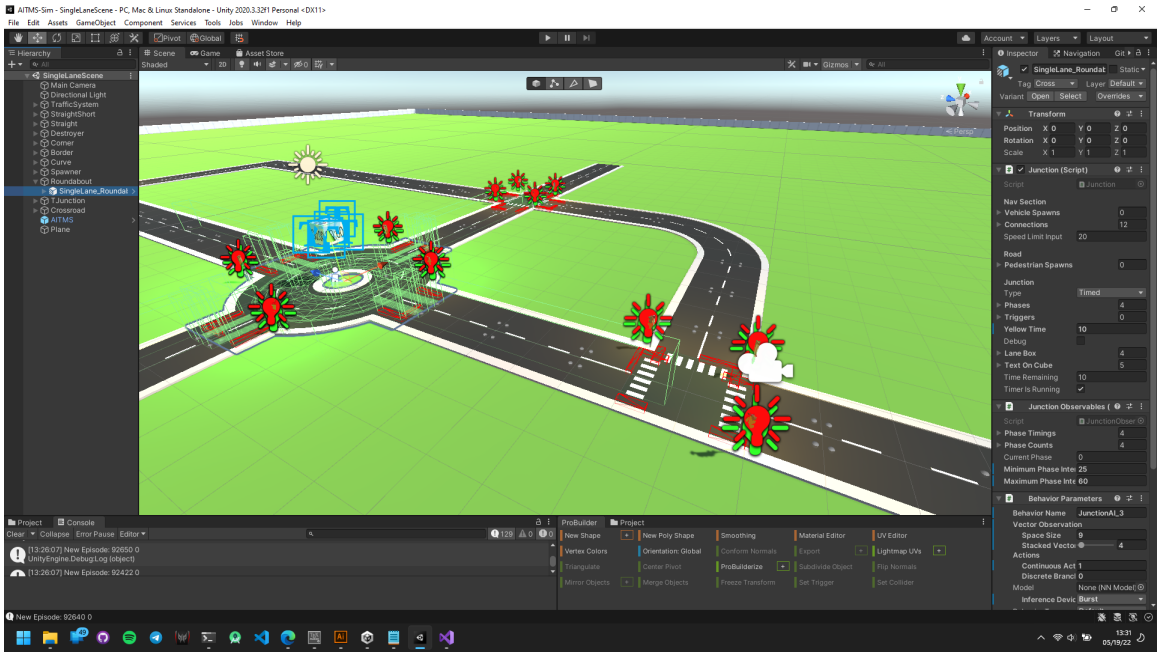

Figure 7.2: AITMS Scene Creation Screen

Figure 7.3: AITMS Simulation Screen

Figure 7.4: AITMS Simulation Screen (Intersection)

Figure 7.5: AITMS Simulation Screen (Roundabout)

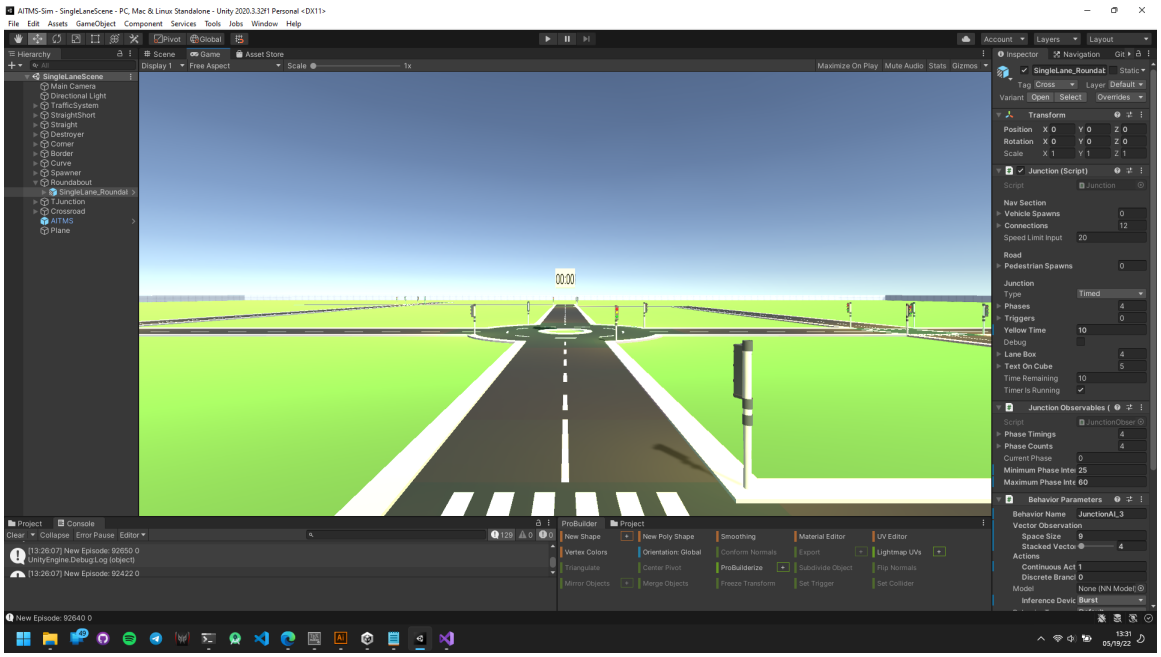

Figure 7.6: AITMS Game View Screen

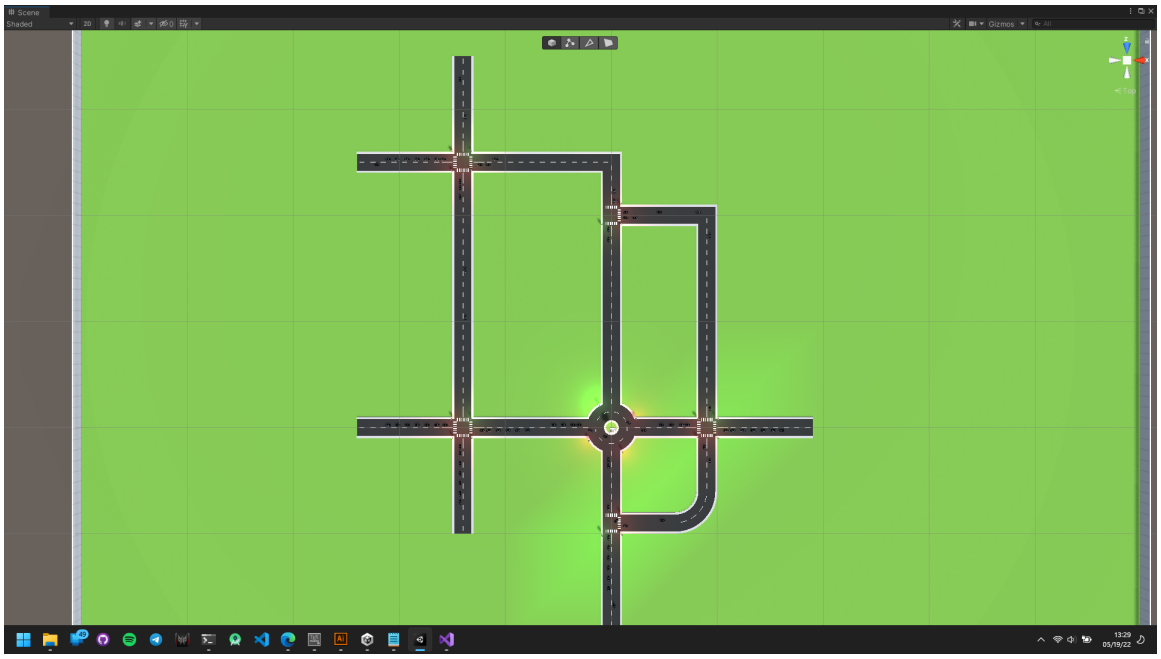

Figure 7.7: AITMS Map Viewer

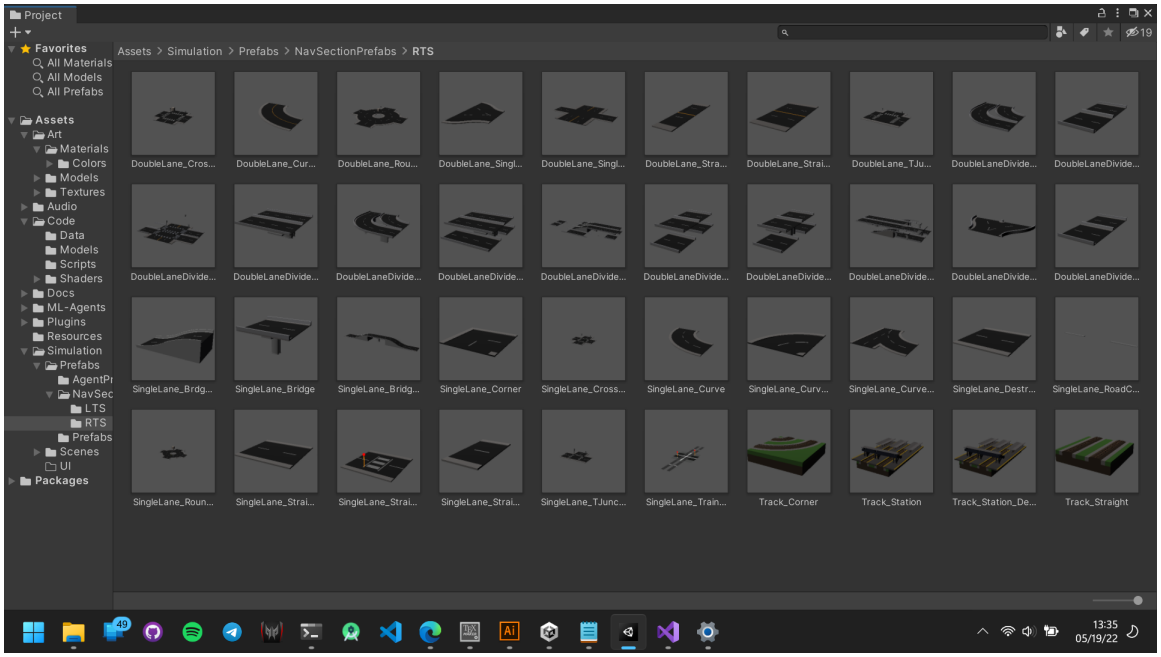

Figure 7.8: AITMS NavSection Prefabs

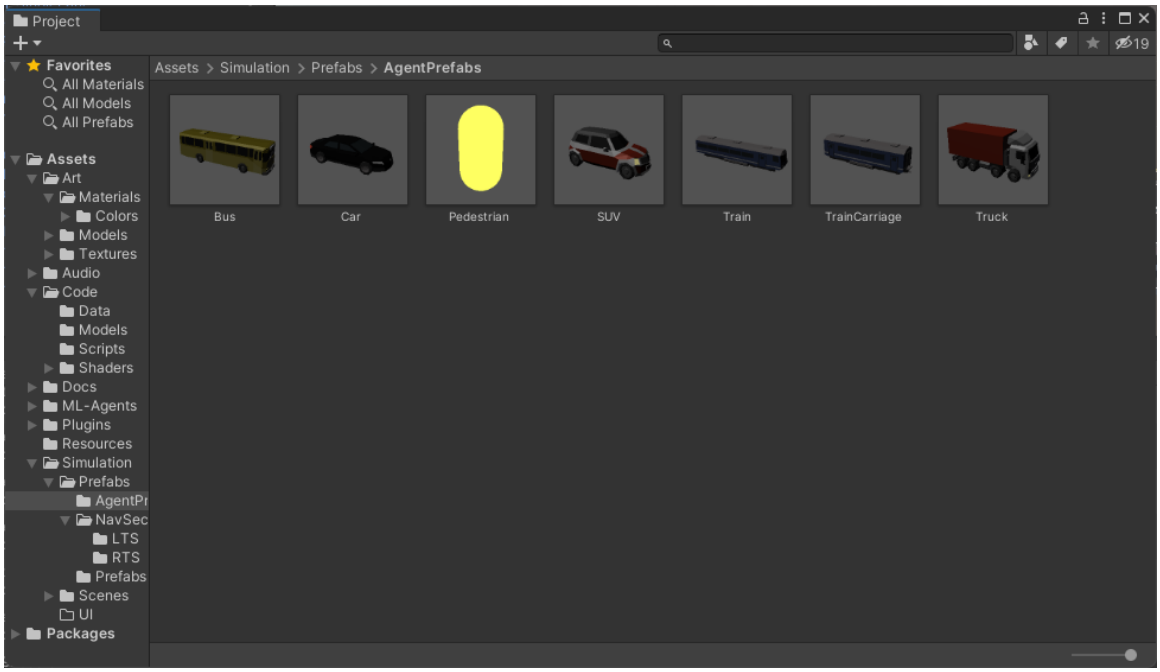

Figure 7.9: AITMS Agent Prefabs

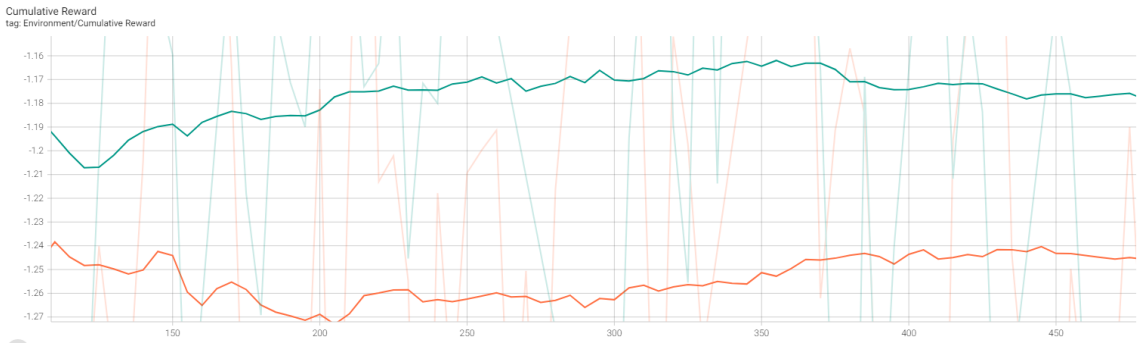

Figure 7.10: AITMS Training Progress Graph (Rewards vs Steps)

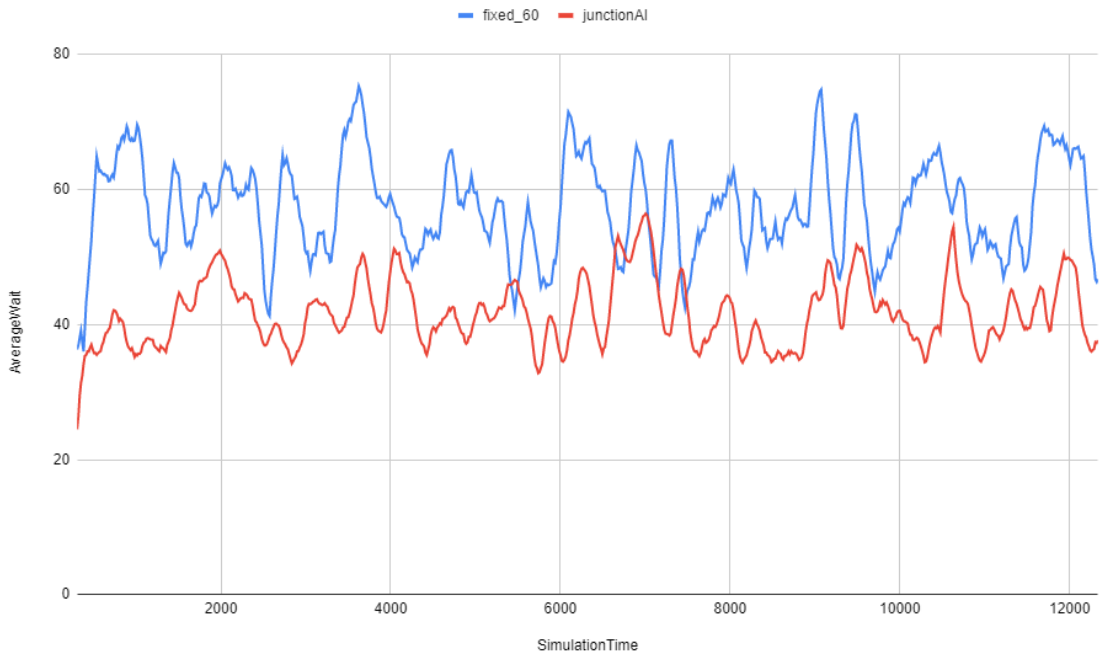

Figure 7.11: AITMS Training Result Graph (AvgWait vs SimulationTime) Showing Approximately 27 % Improvement

7.7 Summary

In this chapter we discussed the implementation details of the Artificially Intelligent Traffic Management System and also the implementation of various features included in the application. We also saw the results in the form of snapshots of the AITMS.

Chapter 8

Testing

This chapter includes the details of Formal Technical Review meetings and describes the process carried during the review process. It also includes the Test Plan adopted for testing the Artificially Intelligent Traffic Management System and Application.

8.1 Formal Technical Review

Formal Technical Reviews and Inspections of documents or software are performed to identify and remove defects. The Formal Technical Review of our project was carried at regular intervals in the form of stand-up meetings and brainstorming sessions conducted. The process included verification of the checklist which was developed for the review process ,the code review checklist template is as follows:

- **Does the code conform to Hungarian Notations?**
- **Is the code well-structured , consistent in style and consistently formatted?**
- **Are all variables properly defined with meaningful,consistent and clear names?**
- **Are there any redundant or unused variables?**
- **Does the code consist of comments ?**
- **Is the code error free?**

8.2 Test Plan

Sr. no.	Module being Tested	Expected Result	Actual Result	Verdict
1	Road assets designing	3D assets of road in Digital Asset Exchange(.dae) file format	3D assets were created using blender in .dae file format	PASS
2	Core Traffic System	3D assets of road in Digital Asset Exchange(.dae) file format	3D assets were created using blender in .dae file format	PASS
3	NavMesh Environtment Setup and Testing	Test if NavMesh package is working properly	NavMesh package working properly	PASS
4	NavConnection implementation to guide vehicle on road	If NavConnection module used to guide vehicle along road and intersections is working as per expectations	NavConnection module working properly and tested	PASS
5	NavSection prefabs	NavSection prefabs which can be used to create maps in sumulation is working properly and aligning to the grid perfectly	NavSections preafabs for common road and intersections working as per expectations	PASS
6	Vehicle(NavAgent) prefabs	NavAgents/Vehicles have realistic physics	NavAgents/Vehicles have physics as per expectations	PASS
7	Road and Juntion Scripts	road and juntion scripts are followed by NavAgent	NavAgents following road and junction scripts as per expectations	PASS

8	Simple Timed Traffic System	Implement a simple timed traffic system core model to navigate traffic, also to bake the NavMesh on runtime/pre-runtime and to manage vehicle pool	Implemented a simple traffic system using scripting and attached it to the relevant object in scene	PASS
9	Map for simulation	If map can be created as per the need of traffic system	Map can be created using NavSection prefabs and is following traffic rules defined in traffic system	PASS
10	Camera Module	Camera Module is working properly	Camera Module is working as per expectations	PASS
11	Junction Observable Module	Junction Observable Module observing junctions properly and collecting data to forward to RL Agents	Junction Observable Module observing junction properly and environment as per expectations and ready to forward data to RL Agents	PASS
12	RL Agents	Test if RL Agents interface is connected with Junction Observable module	RL Agents interface is linked properly with Junction Observable module and is ready to train RL models	PASS
13	Verification of Reward Mechanism	Accurate calculation for reward for given state and action	Rewards are calculated accurately	PASS

14	Vehicle Crash Resolution	In sumulation vehicle should not crash into eachother	Slight crashing between speeding vehicles is observed	FAIL

Table 8.1: Test Plan for AITMS

8.3 Summary

In this chapter we have described the formal technical reviews and the outcome of those. We have described the Test Plan which was successfully carried out at regular development phases.

Chapter 9

Technical Specifications

In this chapter discuss the hardware and software requirements of the proposed Artificially Intelligent Traffic Management System.

9.1 Advantages

Following are some more advantages of Artificially Intelligent Traffic Management System:

- Using defined prefabs, roadmaps can be easily be recreated in simulation.
- Different reward mechanisms can be experimented with in order to train smart signals.
- Simulation and training processes combined extends the use of AITMS to that of a generalized platform for traffic simulation and smart signal training.
- ML Agents library supports the training the model using different types of reinforcement learning trainers such as PPO and SAC. It also includes the option to create recurrent neural networks.
- Underlying technology stack is platform-independent, hence AITMS software (simulation and models) is platform-independent.

9.2 Limitations

- Smart signals can take a long time to train depending on the size of the map, variations in vehicle flow and the type of reward mechanism.
- Larger maps or spawning too many vehicles require more computational power and can cause lag. Hence larger maps with high vehicle density are not feasible to simulate on low-end devices.
- Roads in real life can have shapes that are not included in the set of prefabs (presets). In order, to simulate such roads, their prefabs would have to be created separately.
- Inherent randomness in the movement of vehicles often causes them to crash, this is often seen when phase time is given an insufficiently small value. Traffic jams caused by this are handled by deleting vehicles that have been stationary beyond a certain threshold duration.

9.3 Applications

The Artificially Intelligent Traffic Management System can be used in following areas:

- Help in planning roadways in smart cities by providing insights on traffic flow and how it can be optimized.
- As a platform for simulating traffic and experimenting with smart signal training using various types of models and reward mechanisms.
- Foundation for a more advanced smart signal training platforms to help implementation of smart signals using AI in real life.

9.4 Hardware Requirements

- Intel Core i5 8300H or equivalent/higher
- 16GB RAM for application development
- Min. 16 GB Space in Hard Disk

9.5 Software Requirements

- Visual Studio 2019
- Unity 2020.3.32f
- Python 3.7 or higher
- YOLO 3.0 or higher
- TensorFlow 2.0 or higher
- PyTorch 1.8.1 or higher
- Blender 3.0 or higher
- GIMP and Photoshop

9.6 Summary

This chapter discusses the various hardware and software requirements of the project.

Chapter 10

Future Scope

Scalability is an essential feature for traffic management systems in current age. As the number of vehicles and roads increases for a particular locality, it's traffic management system must be able to handle the changes in maps and traffic flow. Keeping this in mind, the proposed system provides the following future possibilities:

- The ability to connect multiple AITMS will greatly improve scalabbility and optimization. If nieghboring solutions can work together, providing an additional layer of optimization using their combined data becomes possible.
- Improving on library of optimization algorithms. Since, different algorithms may suit different maps, having a variety of algorithms in the toolkit becomes helpful.
- Integration of additional functionalities such as incident detection, manual control, etc.

Chapter 11

Conclusion

AITMS provides a platform for simulating traffic flow and training smart signals using reinforcement learning models which are fed inputs in the form of state observations from various traffic intersections in the simulation. The aim of training is to maximize a reward function which penalizes the model based on average waiting time and vehicle count. Hence, a successfully trained model is able to reduce unnecessary waiting time and at the same time provide just enough green signal time to allow maximum waiting vehicles to pass through. The proposed system for smart traffic signals is able to successfully minimize congestion in simple road maps consisting of 4-5 intersections by approximately upto 25%. The various scripts were made keeping reusability in mind to allow the users to program and experiment with different reward functions.

A wide variety of traffic component prefabs (or presets) with generalized parameters help in replicating characteristics of traffic flow in real life. This helps the simulation provide information valuable to planning road maps since areas prone to congestion and slow traffic flow become easily observable.

AI based smart signals provide sufficient scope for scalability since reduction in congestion helps in dealing with increasing number of vehicles. AITMS is designed to use inputs based on inferences which can be gathered from existing CCTV cameras by using object detection algorithms like YOLO on live footage. This maintains feasibility and eliminates the need to install new sensors.

Furthermore, collection of CCTV footage and traffic data at intersections helps in generating statistical reports on congestion and provides future scope for additional features such as incident detection and detecting traffic violations.

Appendix A

Glossary

- **AITMS:** We have code named our project Artificially Intelligent Traffic Management System as AITMS.
- **AI:** Artificially Intelligent
- **ATCS:** Adaptive Traffic Control System
- **ML:** Machine Learning
- **OPAC:** Optimized Policies for Adaptive Control
- **PPO:** Proximal Policy Optimization
- **RHODES:** Realtime Hierarchical Optimized Distributed Effective System
- **RL:** Reinforcement Learning
- **SAC:** Soft Actor-Critic
- **SCATS:** Sydney Co-ordinated Adaptive Traffic System
- **SCOOT:** Split Cycle Offset Optimization Technique
- **UML:** Unified Modeling Language
- **YOLO:** You Only Look Once

Bibliography

[1] *You Only Look Once: Unified, Real-Time Object Detection*; Joseph Redmon, Santosh Divvala, Ross Girshick and Ali Farhadi, 2016.

[2] *Traffic Congestion Detection from Camera Images using Deep Convolution Neural Networks*; Pranamesh Chakraborty, Yaw Okyere, Subhadipto Poddar, Vesal Ahsani, Anuj Sharma and Soumik Sarkar. In Transportation Research Record Journal of the Transportation Research Board, June 2018. DOI: 10.1177/0361198118777631

[3] *Smart Control of Traffic Light Using Artificial Intelligence*; M. M. Gandhi, D. S. Solanki, R. S. Daptardar and N. S. Baloorkar. 2020 5th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE), 2020, pp. 1-6, doi: 10.1109/ICRAIE51050.2020.9358334.

[4] *Comparison of Current Practical Adaptive Traffic Control Systems.*; Hongyun Chen & Jian Lu. (2010). 1611-1619. 10.1061/41127(382)176.

[5] *Optimizing Networks of Traffic Signals in Real Time - The SCOOT Method.*;D I Robertson and R D Bretherton. IEEE Transactions on Vehicular Technology, 1991.

[6] *The sydney coordinated adaptive traffic (scat) system philosophy and benefits,*;A G Sims and K W Dobinson. 1980.

[7] *Comparison of PPO and SAC Algorithms Towards Decision Making Strategies for Collision Avoidance Among Multiple Autonomous Vehicles*A. J. M. Muzahid, S. F. Kamarulzaman and M. A. Rahman, 2021 International Conference on Software Engineering & Computer Systems and 4th International Conference on Computational Science and Information Management (ICSECS-ICOCSIM), 2021, pp. 200-205, doi: 10.1109/ICSECS52883.2021.00043.

[8] *Li, Zhenning & Xu, Cheng-Zhong & Zhang, Guohui. (2021). A Deep Reinforcement Learning Approach for Traffic Signal Control Optimization.*