**A**

**MINI PROJECT REPORT**

**on**

# SMART TRAFFIC SIGNAL USING MACHINE LEARNING FOR URBAN CONGESTION CONTROL

**Submitted to the Faculty of Engineering and Technology**

**B. Tech - VI Semester**

By


**NARENDAR BOMMERA**

**B23CN133L**


Under the Guidance of
**L.SUJITH KUMAR**
**Assistant Professor**

## Department of Computer Science and Engineering (NETWORKS)

**Kakatiya Institute of Technology &**

**Science (An Autonomous Institute under**

**Kakatiya University) Warangal – Telangana**

**2024-2025**

NAAC - 'A' Grade Institute (CGPA : 3.21) ISO 9001:2015 NIRF - 2020 Rank Band : 201-250

**KAKATIYA INSTITUTE OF TECHNOLOGY & SCIENCE**

Opp : Yerragattu Gutta, Hasanparthy (Mandal), WARANGAL - 506 015, Telangana, INDIA.

काकतीय प्रैद्योगिकी एवं विज्ञान संस्थान, वरंगल - ५०६ ०१५ तेलंगाना, भारत

కాకతీయ సాంకేతిక విజ్ఞాన శాస్త్ర విద్యాలయం, వరంగల్ - ౫౦౬ ౦౧౫ తెలంగాణ, భారతదేశము

Estd-1980

KITSW

*(An Autonomous Institute under Kakatiya University, Warangal)*

*(Approved by AICTE, New Delhi; Recognised by UGC under 2(f) & 12(B); Sponsored by EKASILA EDUCATION SOCIETY)*

website: www.kitsw.ac.in E-mail: principal@kitsw.ac.in ✆ : +91 9392055211, +91 7382564888

# CERTIFICATE

This is to certify that **NARENDAR BOMMERA** bearing Roll No. **B23CN133L** of the VI Semester B.Tech. Computer Science and Engineering (NETWORKS) has satisfactorily completed the Mini Project dissertation work entitled, **"SMART TRAFFIC SIGNAL USING MACHINE LEARNING FOR URBAN CONGESTION CONTROL",** in partial fulfillment of the requirements of B. Tech Degree during this academic year 2024-2025.

**Supervisor**
L.SUJITH KUMAR
**Assistant Professor**

**Coordinator**
V.SRINIVAS
**Assistant Professor**

**Convenor**
Dr. B. V. PRANAY KUMAR
**Associate Professor**

**Head of the Department**
Dr .V. SHANKAR
**Professor**

# ACKNOWLEDGEMENT

I extend my sincere heartfelt thanks to our esteemed counselor/guide **L.SUJITH KUMAR, Assistant Professor** for him exemplary guidance, monitoring and consistent encouragement throughout the course at crucial junctures and for showing us the right way.

It is a great privilege for me to here by deep sense of gratitude towards Mini- Project Coordinator **V.SRINIVAS, Assistant Professor** for guiding and permitting me in successful completion of my work on time.

I am grateful to respected Mini-Project Convenor **Dr. B. V. PRANAY KUMAR, Assistant Professor** for guiding and permitting me to utilize all the necessary facilities of the Institute.

I would like to extend thanks to **Dr .V. SHANKAR, Professor & Head of the Department** for allowing me to use the facilities available.

I express my immense delight to **Assistant professor L.SUJITH KUMAR** for his kind cooperation and elderly suggestions throughout the Mini-Project work. I am very much thankful for issuing similarity report through Turnitin Anti Plagiarism Software.

I take this opportunity to express my sincere and heartfelt gratitude to the honorable **Principal Prof. K. ASHOKA REDDY** of our institute for granting me permission to do the Mini- Project.I am very happy in expressing my sincere thanks to faculty & staff, Department of Computer Science and Engineering, friends and family for the support and encouragement that they have given us during the Mini-Project.

**NARENDAR BOMMERA**

**B23CN133L**

# ABSTRACT

SMART TRAFFIC SIGNAL USING MACHINE LEARNING FOR URBAN CONGESTION CONTROL key to efficient traffic management in smart cities. While RADAR or inductive loop based systems are regularly used for the task in the developed world, financial and infrastructure constraints precludes use of sophisticated and expensive sensors in the developing countries.

The on going project provides the techniques to minimize traffic in the city of Bangalore using the Reinforcement Learning. The project was aimed at successful deployment of RL model using Feed Forward network and LSTM(Long Short Term Memory) on a single traffic junction and optimize the network. The current agent give a far lesser delay time than the FST -60( Fixed standard time of 60 seconds via round roubin technique to each section of the junction). After the current optimization of a single juntion using RL model we plan to further expand to it to multiple junctions using Mulit agent Reinforcement L

## TABLE OF CONTENTS

# CHAPTER 1

## INTRODUCTION

The number of vehicles on the road are increasing day by day so it is important to manage the traffic flow efficiently in order to utilize the existing road capacity in the best way possible. Developing a smart traffic management system to optimize traffic flow, reduce congestion, while minimizing the travel time and maximizing mobility. Installation of traffic signals can actually cause a deterioration in overall safety of intersections. Time traffic signals can cause a situation of deadlock. Metro cities and many majorly populated cities have traffic signals at very short distances which prevent the smooth flow of traffic. Severe traffic can cause phantom traffic jams. The present automated traffic control systems work on time-based algorithms. Each lane is allotted a fixed time for traffic to clear off, the times may be equal for all lanes or based on the average vehicle density.

### 1.1 OVERVIEW OF THE CURRENT INVESTIGATION

We successfully implemented a Reinforcement Learning (RL) algorithm for managing traffic at a single junction. The model has been tested extensively using various state and action spaces along with a well-defined reward function, all aimed at minimizing traffic delay. Compared to the traditional FST-60 approach (which gives a fixed 60-second green signal to each section), our RL agent consistently achieves reduced wait times and smoother traffic flow.

Here are some key highlights and accomplishments from deploying the RL model:

1. **Effective Communication Setup:** Established seamless communication between the Aimsun traffic simulator and an external Python-based control system.

2. **Custom Environment Creation:** Since Aimsun doesn't natively support Python integration, we developed a custom *Environment* class to bridge the gap. This was crucial for controlling the simulation externally.

3. **Real-Time Data Extraction:** Successfully fetched key traffic metrics such as density, queue length, and delay time for each direction of the junction. These data points were essential for training the RL model.

4. **Realistic Vehicle Modelling:** Integrated Indian-specific vehicles like auto-rickshaws and motorbikes into the 3D simulation to better reflect real-world conditions.

5. **RL Integration with Environment:** Built the RL training pipeline around the Environment class, following the structure of OpenAI Gym environments. This allowed for easier testing and generalization.

6. **Model Training Across Junctions:** Trained the RL model on the selected junction while feeding it traffic patterns from other FST-based junctions. This helped the model generalize and perform better under diverse conditions.

7. **Exploration of Multiple State-Action Spaces:** Trained several models with different configurations to identify the optimal setup for minimizing traffic delay.

## 1.2 LITERATURE REVIEW

- Traffic congestion is a pressing urban issue with negative economic, environmental, and societal impacts. This literature review aims to explore and evaluate diverse strategies to address traffic congestion.

- Causes and Effects: Traffic congestion stems from factors like population growth, limited road capacity, and car dependency.

- Traffic lights play a pivotal role in managing intersections and regulating traffic. They facilitate the orderly movement of vehicles, pedestrians, and cyclists by assigning right-of-way and controlling signal timing.

- Numerous cities have adopted adaptive signal control systems and coordinated signal timing strategies. Successful implementations have demonstrated reduced travel times, enhanced traffic flow, and minimized congestion.

- Benefits and Challenges: Benefits of traffic light solutions include improved traffic flow, reduced congestion, and potential fuel savings. However, challenges include initial implementation costs, maintenance, and potential disruptions during deployment.

- In our project, we have to select at least a 5MP day and night IP camera for capturing real-time traffic data at key traffic junctions. The choice of camera is a critical component as it directly affects the accuracy and efficiency of our traffic management system.

- The efficiency of our traffic management system also depends on the computer system used for data processing and control logic execution. We need at least an Intel Celeron or AMD A9 processer with good network connectivity. The selected computer system should meet or exceed these specifications to ensure smooth video processing, real-time traffic analysis, and adaptive signal control.

## 1.3 MOTIVATION

The motivation behind embarking on this project is multi-faceted, stemming from both personal and societal aspirations. As technology continues to shape our world, it becomes imperative to leverage its power to address pressing urban challenges like traffic problems. Here are some key reasons driving our enthusiasm for this project.

- We're eager to delve into the world of computer vision and object detection, to address urban challenges like traffic issues.
- Real-world Application of Technical Skills: We aim to apply our theoretical knowledge in practical ways in the traffic management to make a meaningful impact on the lives of city residents.

## 1.4 PROBLEM DEFINITION

- Developing a smart traffic management system to optimize traffic flow, reduce congestion, while minimizing the travel time and maximizing mobility. The problem at hand is to design and implement a solution that effectively reduces traffic congestion. The solution should focus on minimizing congestion-related delays, improving travel times, reducing environmental impact, and enhancing overall urban mobility.

## 1.4 OBJECTIVES

- The main objective of this project is to design a traffic light controller based on Computer Vision that can adapt to the current traffic situation. Our proposed system aims to use live video feed from the CCTV cameras at traffic junctions for real- time traffic density calculation by detecting the vehicles at the signal and setting the green signal time accordingly.

- It will enhance the efficiency of the transportation system by optimizing traffic management, reducing bottlenecks, and ensuring smoother coordination between various transportation modes.
- Increases the safety for pedestrians, cyclists, and drivers by implementing measures that reduce accidents, improve visibility and prioritize pedestrian-friendly infrastructure.

## 1.5 EXPECTED OUTCOMES

- Engaging in a project focused on solutions for traffic congestion using traffic lights can yield a range of expected outcomes.
- Improved Traffic Flow: Implementation of optimized traffic light strategies can lead to smoother traffic flow, reduced stop-and-go patterns, and decreased congestion at intersections.
- Reduced Travel Time: By minimizing waiting times at traffic lights, commuters experience reduced travel time, leading to enhanced efficiency in daily transportation.
- Enhanced Safety: Well-coordinated traffic lights contribute to safer road conditions by reducing abrupt stops and minimizing the risk of collisions at intersections.
- Technological Innovation: Implementing adaptive traffic signal control and intelligent transportation systems showcases the practical application of cutting-edge technologies.

# CHAPTER 2

## PROPOSED METHODOLOGY

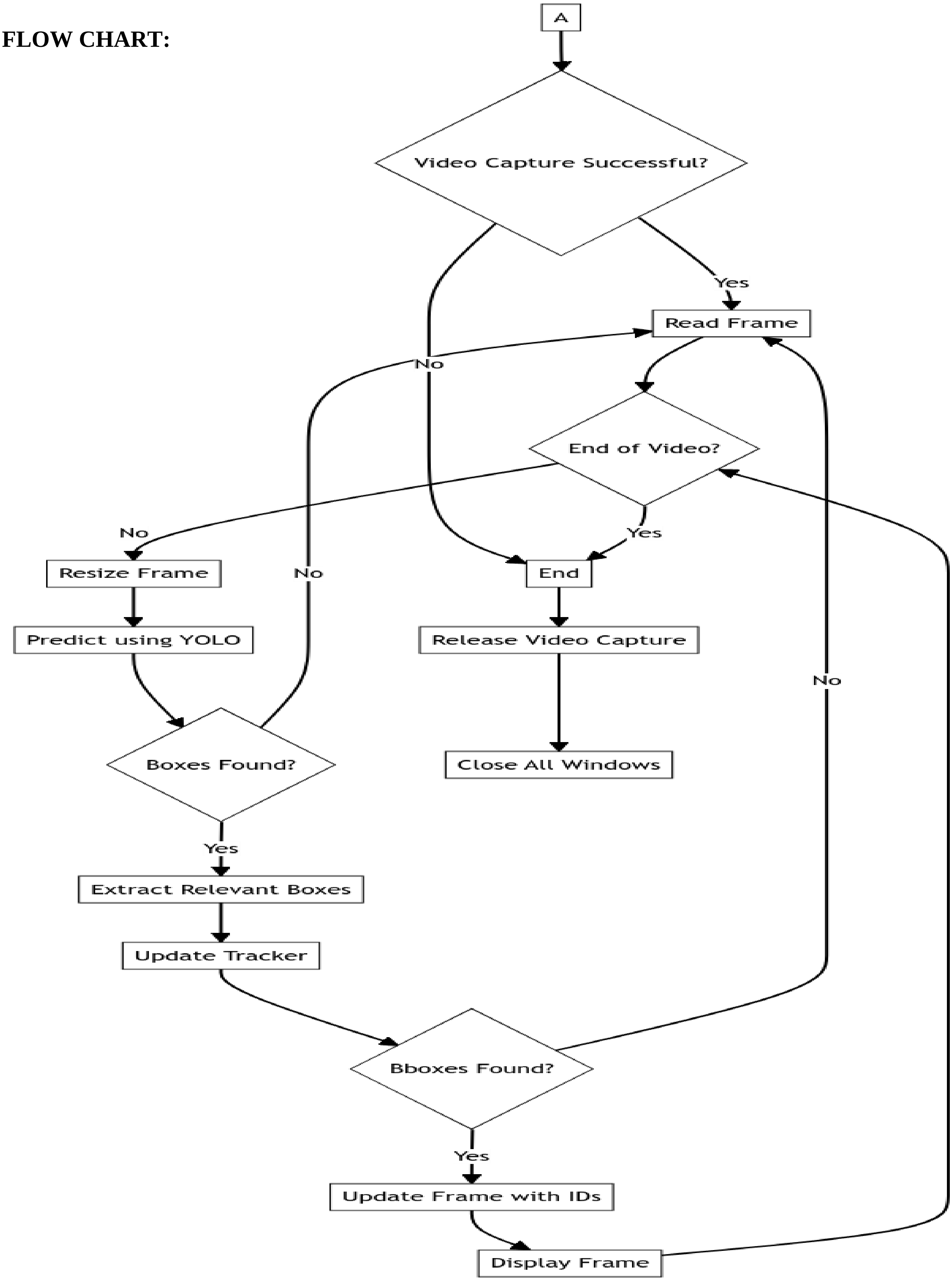
In this chapter, we outline our approach to tackle the challenges posed by traffic congestion and signal control. Our methodology focuses on developing an adaptive traffic signal system that responds to real-time traffic conditions, promoting efficient traffic flow and congestion reduction.

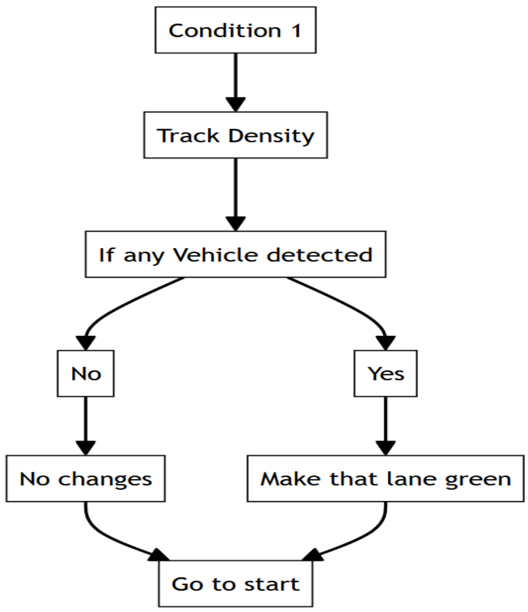
- Problem Background and Rationale: Traffic congestion is a persistent issue in urban areas due to fixed signal timings that fail to accommodate varying traffic patterns. Conventional signal timing methods result in traffic jams, delays, accidents, and increased pollution. To address these issues, a dynamic approach to signal control is imperative.
- Proposed System Overview: Our approach involves strategically deploying CCTV cameras at key traffic junctions. These cameras capture real-time snapshots of traffic scenarios, which are then subjected to advanced Image Processing and Computer Vision techniques. These methods extract crucial data about traffic density, allowing us to gauge the current traffic situation accurately.
- Traffic Flow Analysis: By analyzing the data obtained from CCTV cameras, we can perform instant traffic flow analysis. This entails identifying lanes with high and low traffic densities. This analysis forms the basis for determining how much green signal time should be allocated to each direction.
- Dynamic Signal Timing Calculation: Leveraging the insights gained from traffic flow analysis, we compute optimal green signal timings dynamically. The direction with higher traffic density receives a longer green signal duration compared to directions with lighter traffic. This adaptability aims to alleviate congestion and enhance traffic flow efficiency.

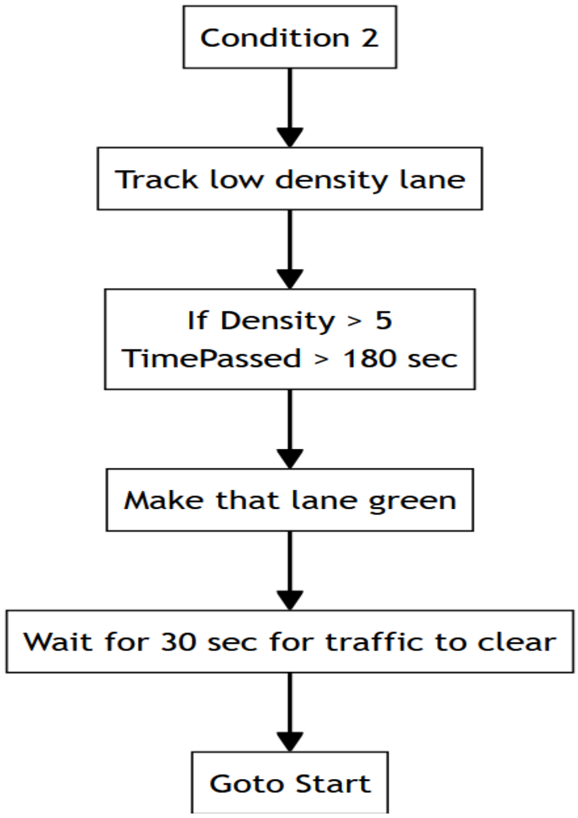
Implementation of Control Logic: Our approach includes integrating the computed signal timings with the actual traffic signal hardware. This integration is achieved through microcontrollers or similar technology. By doing so, we enable real-time communication between our dynamic calculations and the physical operation of traffic signals, ensuring synchronization.

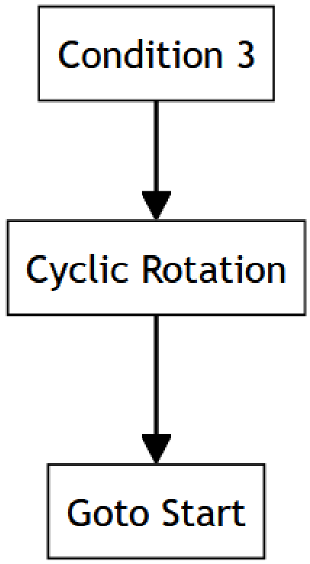
### 2.1 ALGORITHMS

### FLOW CHART:




Condition 1
Track Density
If any Vehicle detected
No
Yes
No changes
Make that lane green
Go to start


Condition 2
Track low density lane
If Density > 5
TimePassed > 180 sec
Make that lane green
Wait for 30 sec for traffic to clear
Goto Start


Condition 3
Cyclic Rotation
Goto Start

**Data:** Traffic density data from YOLO, FTP connection parameters, and traffic management logic
**Result:** Real-time traffic management based on density data

```
Initialize YOLO for vehicle detection and density calculation;
Initialize FTP connection parameters for data transfer;
while true do
    Detect and calculate traffic density using YOLO;
    if Traffic density data changes then
        Transmit updated density data to the remote computer via FTP;
    end
    if Receiving computer detects new density data then
        Receive the data via FTP;
        Parse the received data to extract traffic density information;
        Implement traffic management logic based on the density data;
    end
    Execute traffic management actions, such as signal control or road
     closures;
    Monitor and adapt to changing traffic conditions;
    if Termination condition is met then
        Terminate the loop;
    end
end
Close FTP connection and release resources;
```

**Algorithm 4:** Real-time Traffic Management based on Density Data

**Required code For Simulation**

```
import random
import time
import matplotlib.pyplot as plt
VEHICLE_TYPES = ['Car', 'Bike', 'Bus', 'Truck', 'Auto']
VEHICLE_COLORS = {
    'Car': '#8B0000',      # Dark Red
    'Bike': '#000080',     # Navy Blue
    'Bus': '#FF8C00',      # Dark Orange
    'Truck': '#800080',    # Purple
    'Auto': '#008080'      # Teal
}
class Lane:
    def __init__(self, lane_id):
        self.lane_id = lane_id
        self.vehicle_count = 0
        self.vehicle_breakdown = {}
        self.green_time = 0
    def detect_vehicles(self):
        total = random.randint(10, 50)
        remaining = total
        breakdown = {}
        for i, v_type in enumerate(VEHICLE_TYPES):
            if i == len(VEHICLE_TYPES) - 1:
                breakdown[v_type] = remaining
            else:
                max_assign = remaining - (len(VEHICLE_TYPES) - i - 1)
```

```
            count = random.randint(1, max_assign)
            breakdown[v_type] = count
            remaining -= count
        self.vehicle_breakdown = breakdown
        self.vehicle_count = total
        return breakdown
    def calculate_green_time(self):
        self.green_time = self.vehicle_count // 2
        return self.green_time
class TrafficSignal:
    def __init__(self, num_lanes=4):
        self.lanes = [Lane(i) for i in range(num_lanes)]
    def simulate_traffic_cycle(self):
        print("\n====== Traffic Cycle Start ======\n")
        lane_ids = []
        vehicle_counts = []
        green_times = []
        vehicle_types_data = []
        for lane in self.lanes:
            breakdown = lane.detect_vehicles()
            lane_ids.append(f"Lane {lane.lane_id}")
            vehicle_counts.append(lane.vehicle_count)
            green_times.append(lane.calculate_green_time())
                    vehicle_types_data.append([breakdown.get(v_type, 0) for v_type in
VEHICLE_TYPES])
            print(f"--- Lane {lane.lane_id} ---")
            for v_type, count in breakdown.items():
                print(f"  {v_type}: {count}")
```

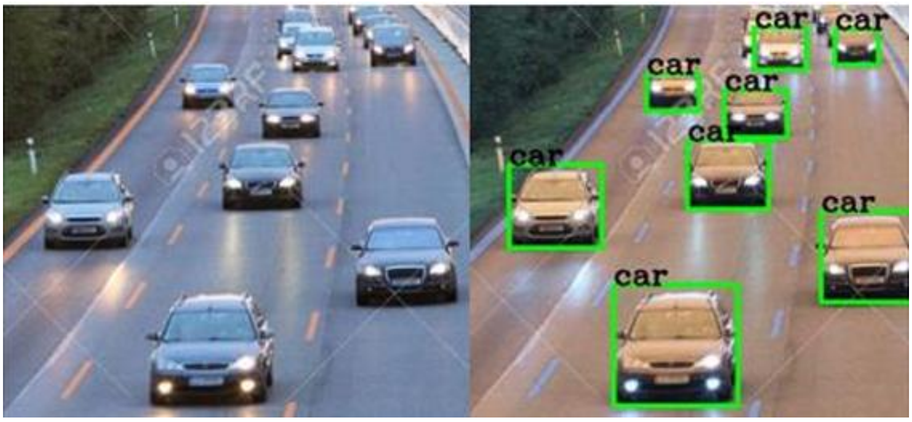
```
            print(f"Total Vehicles: {lane.vehicle_count}")
            print(f"Green Signal Time: {lane.green_time} seconds\n")
        self.plot_graph(lane_ids, vehicle_types_data, green_times)
        for lane in self.lanes:
            print(f"\nLane {lane.lane_id} gets green signal for {lane.green_time} seconds\n")
            self.animate_signal(lane)
    def animate_signal(self, lane):
        for t in range(lane.green_time, 0, -1):
            print(f"Lane {lane.lane_id} Green: {t}s")
            time.sleep(1)
        print(f"Lane {lane.lane_id} turns red.\n")
    def plot_graph(self, lane_ids, vehicle_types_data, green_times):
        x = range(len(lane_ids))
        bar_width = 0.12
        fig, ax = plt.subplots(figsize=(13, 6))
        for i, v_type in enumerate(VEHICLE_TYPES):
            positions = [pos + i * bar_width for pos in x]
            values = [vehicle_types_data[j][i] for j in range(len(x))]
            bars = ax.bar(positions, values, bar_width, label=v_type, color=VEHICLE_COLORS[v_type])


            for bar in bars:
                height = bar.get_height()
                ax.text(bar.get_x() + bar.get_width()/2, height + 0.5, f'{int(height)}', ha='center', fontsize=8, color='black')

        green_pos = [pos + len(VEHICLE_TYPES) * bar_width for pos in x]
```

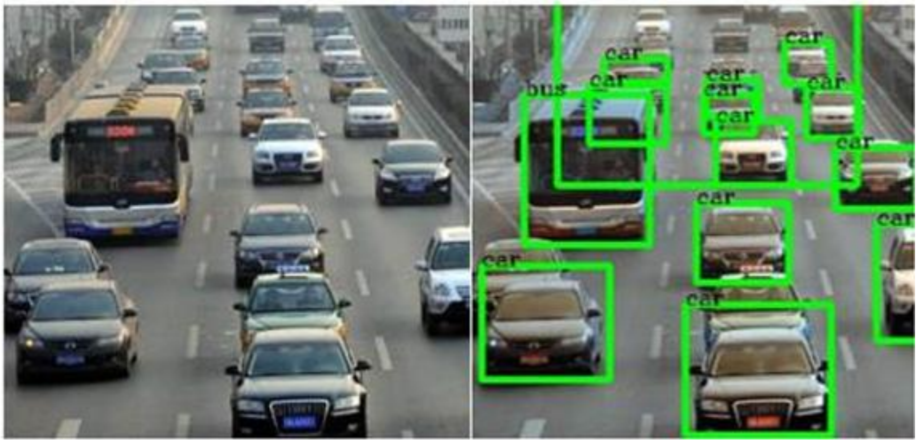
```
        green_bars = ax.bar(green_pos, green_times, bar_width, label='Green Time (s)', color='#00C853')



        for bar in green_bars:
            height = bar.get_height()
            ax.text(bar.get_x() + bar.get_width()/2, height + 0.5, f'{int(height)}', ha='center', fontsize=8, color='black')
        ax.set_xticks([pos + (len(VEHICLE_TYPES) * bar_width) / 2 for pos in x])
        ax.set_xticklabels(lane_ids)
        ax.set_xlabel("Lane")
        ax.set_ylabel("Count / Time (s)")
        ax.set_title("Vehicle Types and Green Signal Time per Lane", fontsize=14, fontweight='bold')
        ax.legend(loc='upper right')
        plt.tight_layout()
        plt.show()
def main():
    traffic_signal = TrafficSignal()
    traffic_signal.simulate_traffic_cycle()
if __name__ == "__main__":
    main()
```

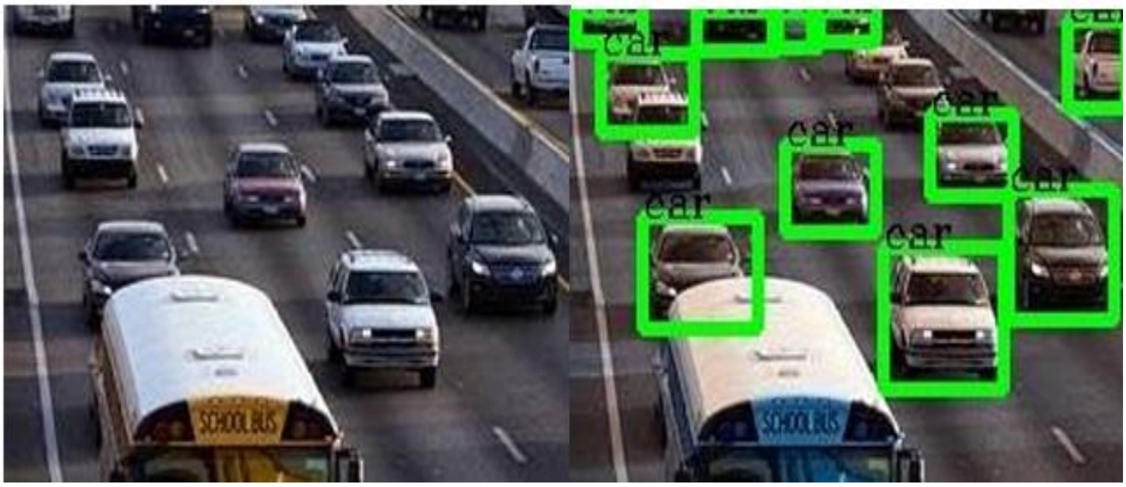
### 2.3 Vehicle Detection Module:

- The proposed system uses YOLO (You only look once) for vehicle detection, which provides the desired accuracy and processing time. A custom YOLO model was trained for vehicle detection, which can detect vehicles of different classes like cars, bikes, heavy vehicles (buses and trucks), and rickshaws.
- The dataset for training the model was prepared by scraping images from google and labelling them manually using Label IMG, a graphical image annotation tool.
- Then the model was trained using the pre-trained weights downloaded from the YOLO website. The configuration of the .cfg file used for training was changed in accordance with the specifications of our model. The number of output neurons in the last layer was set equal to the number of classes the model is supposed to detect by changing the 'classes' variable. In our system, this was 4 viz. Car, Bike, Bus/Truck, and Rickshaw. The number of filters also needs to be changed by the formula 5*(5+number of classes), i.e., 45 in our case.
- After making these configuration changes, the model was trained until the loss was significantly less and no longer seemed to reduce. This marked the end of the training, and the weights were now updated according to our requirements.
- These weights were then imported in code and used for vehicle detection with the help of OpenCV library. A threshold is set as the minimum confidence required for successful detection. After the model is loaded and an image is fed to the model, it gives the result in a JSON format i.e., in the form of key-value pairs, in which labels are keys, and their confidence and coordinates are values. Again, OpenCV can be used to draw the bounding boxes on the images from the labels and coordinates received.

**Following are some images of the output of the Vehicle Detection Module:**






### Switching Algorithm

The Signal Switching Algorithm sets the green signal timer according to traffic density returned by the vehicle detection module, and updates the red signal timers of other signals accordingly. It also switches between the signals cyclically according to the timers.

The algorithm takes the information about the vehicles that were detected from the detection module, as explained in the previous section, as input. This is in JSON format, with the label of the object detected as the key and the confidence and coordinates as the values. This input is then parsed to calculate the total number of vehicles of each class. After this, the green signal time for the signal is calculated and assigned to it, and the red signal times of other signals are adjusted accordingly. The algorithm can be scaled up or down to any number of signals at an intersection.

The following factors were considered while developing the algorithm:

1. The processing time of the algorithm to calculate traffic density and then the green light duration – this decides at what time the image needs to be acquired
2. Number of lanes
3. Total count of vehicles of each class like cars, trucks, motorcycles, etc.
4. Traffic density calculated using the above factors
5. Time added due to lag each vehicle suffers during start-up and the non-linear increase in lag suffered by the vehicles which are at the back
6. The average speed of each class of vehicle when the green light starts i.e. the average time required to cross the signal by each class of vehicle
7. The minimum and maximum time limit for the green light duration - to prevent starvation

## 2.5 Working of the algorithm

times for all other signals of the first cycle and all signals of the subsequent cycles are set by the algorithm. A separate thread is started which handles the detection of vehicles for each direction and the main thread handles the timer of the current signal. When the green light timer of the current signal (or the red light timer of the next green signal) reaches 0 seconds, the detection threads take the snapshot of the next direction. The result is then parsed and the timer of the next green signal is set. All this happens in the background while the main thread is counting down the timer of the current green signal. This allows the assignment of the timer to be seamless and hence prevents any lag When the algorithm is first run, the default time is set for the first signal of the first cycle and the. Once the green timer of the current signal becomes zero, the next signal becomes green for the amount of time set by the algorithm.

The image is captured when the time of the signal that is to turn green next is 0 seconds. This gives the system a total of 5 seconds (equal to value of yellow signal timer) to process the image, to detect the number of vehicles of each class present in the image, calculate the green signal time, and accordingly set the times of this signal as well as the red signal time of the next signal. To find the optimum green signal time based on the number of vehicles of each class at a signal, the average speeds of vehicles at startup and their acceleration times were used, from which an estimate of the average time each class of vehicle takes to cross an intersection was found. The green signal time is then calculated using the formula below.

$$GST = \frac{\sum_{vehicleClass} (NoOfVehicles_{vehicleClass} * AverageTime_{vehicleClass})}{(NoOfLanes + 1)}$$

where:

- GST is green signal time
- noOfVehiclesOfClass is the number of vehicles of each class of vehicle at the signal as detected by the vehicle detection module,
- averageTimeOfClass is the average time the vehicles of that class take to cross an intersection, and
- noOfLanes is the number of lanes at the intersection.

The average time each class of vehicle takes to cross an intersection can be set according to the location, i.e., region-wise, city-wise, locality-wise, or even intersection-wise based on the characteristics of the intersection, to make traffic management more effective. Data from the respective transport authorities can be analyzed for this.

The signals switch in a cyclic fashion and **not** according to the densest direction first. This is in accordance with the current system where the signals turn green one after the other in a fixed pattern and does not need the people to alter their ways or cause any confusion. The order of signals is also the same as the current system, and the yellow signals have been accounted for as well.

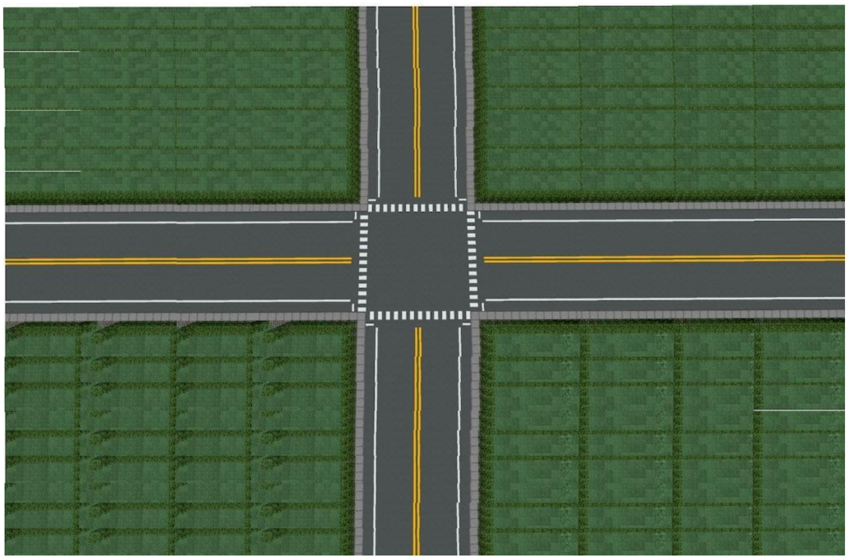
**Order of signals: Red → Green → Yellow → Red**

### 2.6 Simulation Module

A simulation was developed from scratch using Pygame to simulate real-life traffic. It assists in visualizing the system and comparing it with the existing static system. It contains a 4-way intersection with 4 traffic signals. Each signal has a timer on top of it, which shows the time remaining for the signal to switch from green to yellow, yellow to red, or red to green. Each signal also has the number of vehicles that have crossed the intersection displayed beside it. Vehicles such as cars, bikes, buses, trucks, and rickshaws come in from all directions. In order to make the simulation more realistic, some of the vehicles in the rightmost lane turn to cross the intersection. Whether a vehicle will turn or not is also set using random numbers when the vehicle is generated. It also contains a timer that displays the time elapsed since the start of the simulation

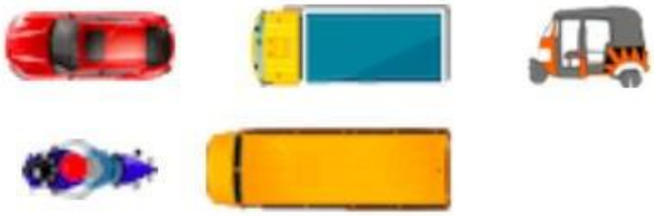
**Key steps in development of simulation:**

1. look an image of a 4-way intersection as background.

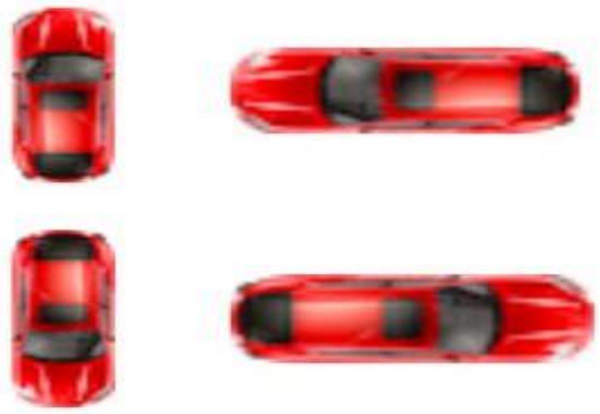


2. Gathered top-view images of car, bike, bus, truck, and rickshaw.

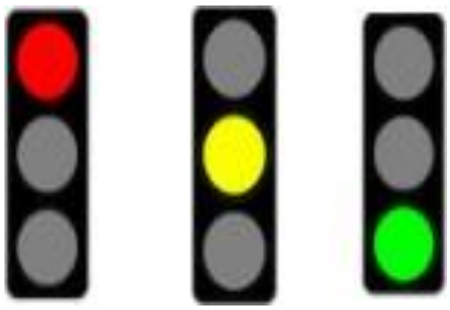
   and Resized them.



3.Rotated them for display along different directions.



4.Gathered images of traffic signals - red, yellow, and green.



1. Code: For rendering the appropriate image of the signal depending on whether it is red, green, or yellow.
2. Code: For displaying the current signal time i.e. the time left for a green signal to turn yellow or a red signal to turn green or a yellow signal to turn red. The green time of the signals is set according to the algorithm, by taking into consideration the number of vehicles at the signal. The red signal times of the other signals are updated accordingly.
3. Generation of vehicles according to direction, lane, vehicle class, and whether it will turn or not all set by random variables. Distribution of vehicles among the 4

directions can be controlled. A new vehicle is generated and added to the simulation after every 0.75 seconds.

4. Code: For how the vehicles move, each class of vehicle has different speed, there is a gap between 2 vehicles, if a car is following a bus, then its speed is reduced so that id does
not crash into the bus.
5. Code: For how they react to traffic signals i.e. stop for yellow and red, move for green. If they have passed the stop line, then continue to move if the signal turns yellow.
6. Code: For displaying the number of vehicles that have crossed the signal.

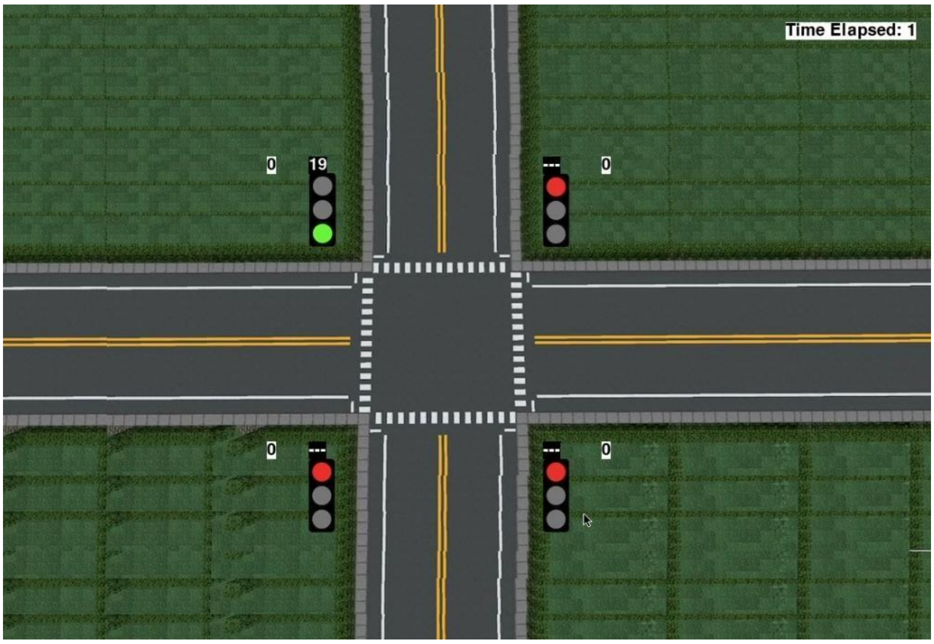
7. Code: For displaying the time elapsed since the start of the simulation.

8. Code: For updating the time elapsed as simulation progresses and exiting when the time elapsed equals the desired simulation time, then printing the data that will be used for comparison and analysis.

9. To make the simulation closer to reality, even though there are just 2 lanes in the image, we add another lane to the left of this which has only bikes, which is generally the case in many cities.

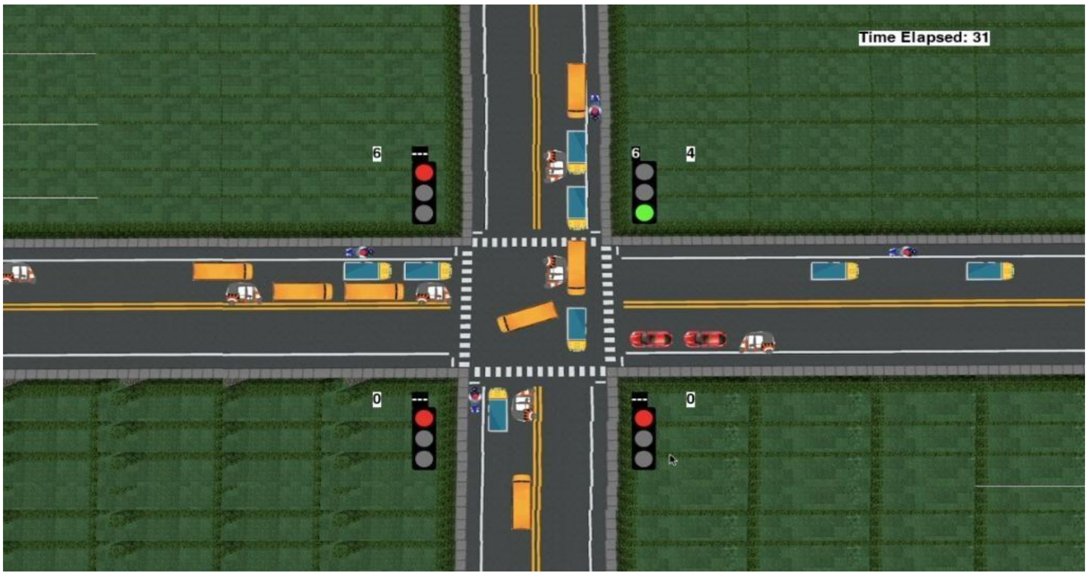
10. Vehicles turning and crossing the intersection in the simulation to make it more realistic.

**Following are some images of the final simulation:**

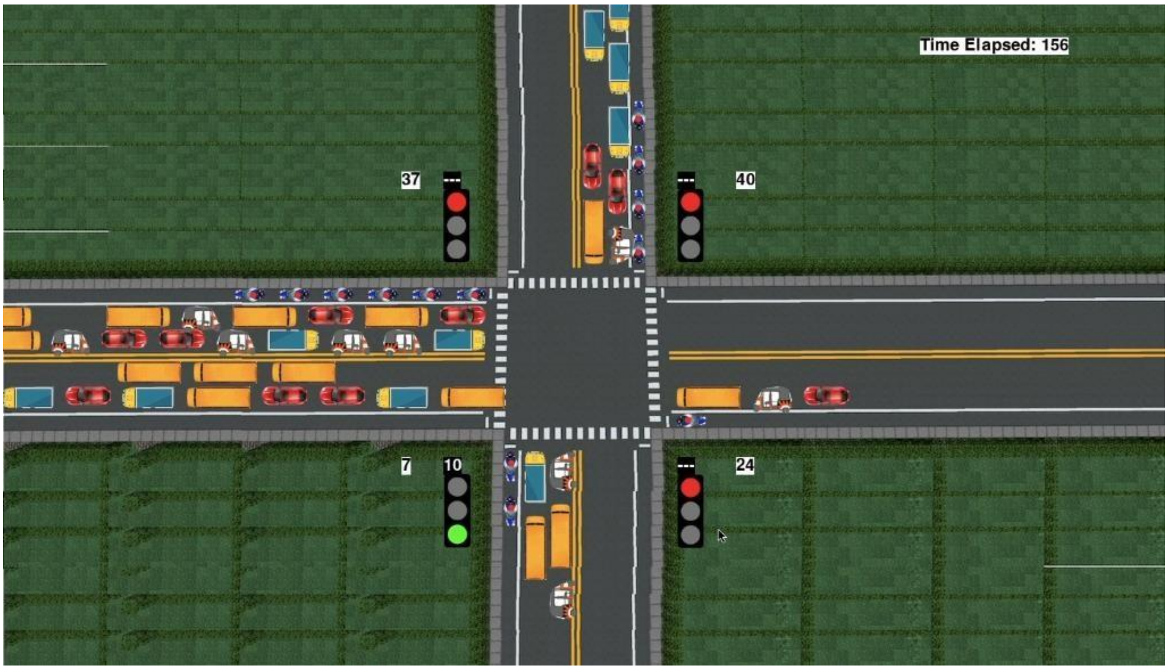



*(i): Simulation just after start showing red and green lights, green signal time counting down from a default of 20 and red time of next signal blank. When the signal is red, we display a blank value till it reaches 10 seconds. The number of vehicles that have crossed can be seen beside the signal, which are all 0 initially. The time elapsed since the start of simulation can be seen on top right.*

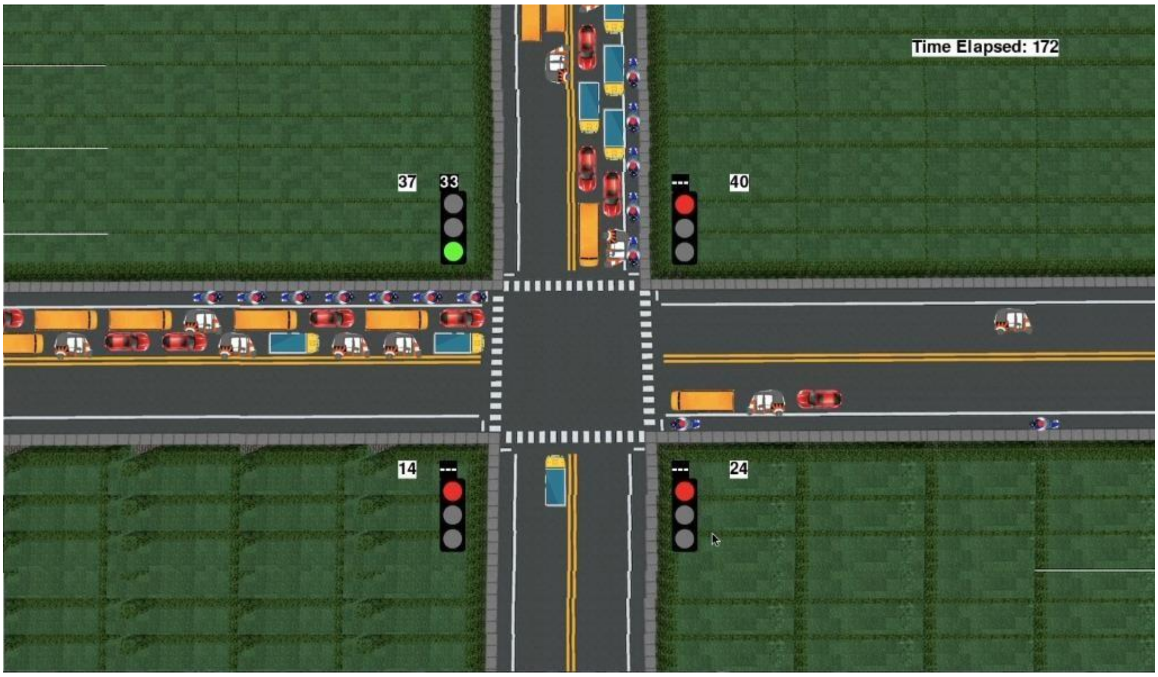

*(ii): Simulation showing yellow light and red time for next signal. When red signal time is less than 10 seconds, we show the countdown timer so that vehicles can start up and be prepared to move once the signal turns green.*




*(iii): Simulation showing vehicles turning*

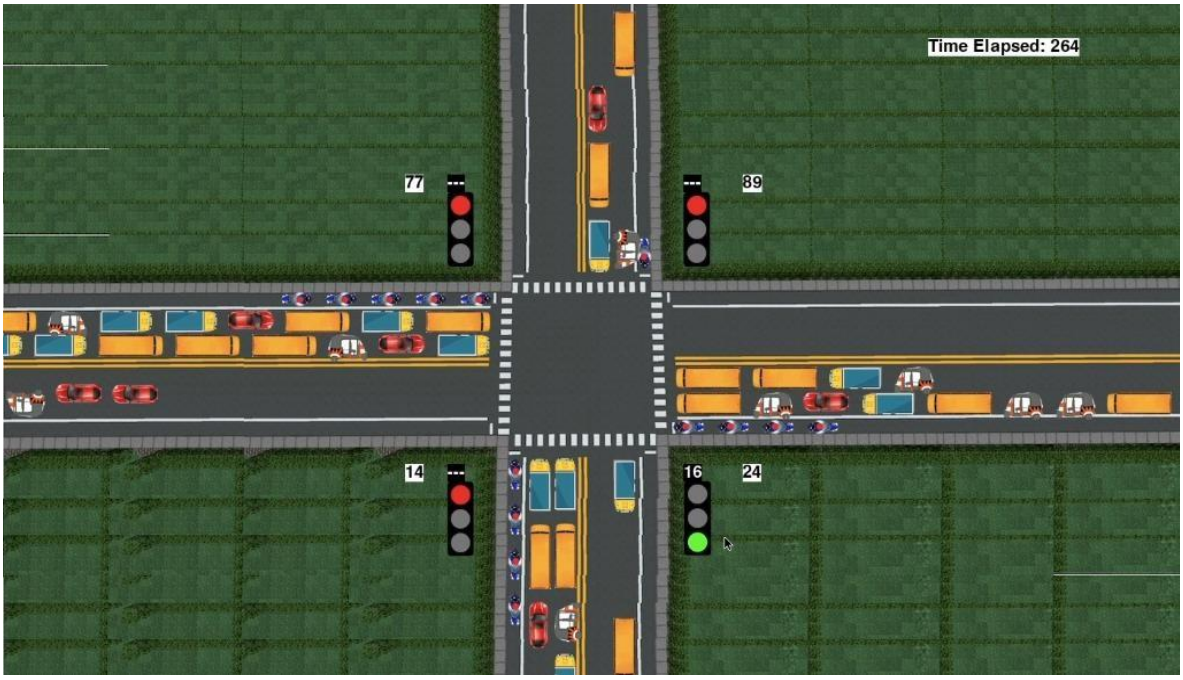



(iv): Simulation showing green time of signal for vehicles moving up set to 10 seconds according to the vehicles in that direction. As we can see, the number of vehicles is quite less here as compared to the other lanes. With the current static system, the green signal time would have been the same for all signals, like 30 seconds. But in this situation, most of this time would have been wasted. But our adaptive system detects that there are only a few vehicles, and sets the green time accordingly, which is 10 seconds in this case.




*(v)Simulation showing green time of signal for vehicles moving right set to 33*

*seconds according to the vehicles in that direction.*




*(vi): Simulation showing green time of signal for vehicles moving left set to 24 seconds according to the vehicles in that direction*

# CHAPTER 3

**Output:**

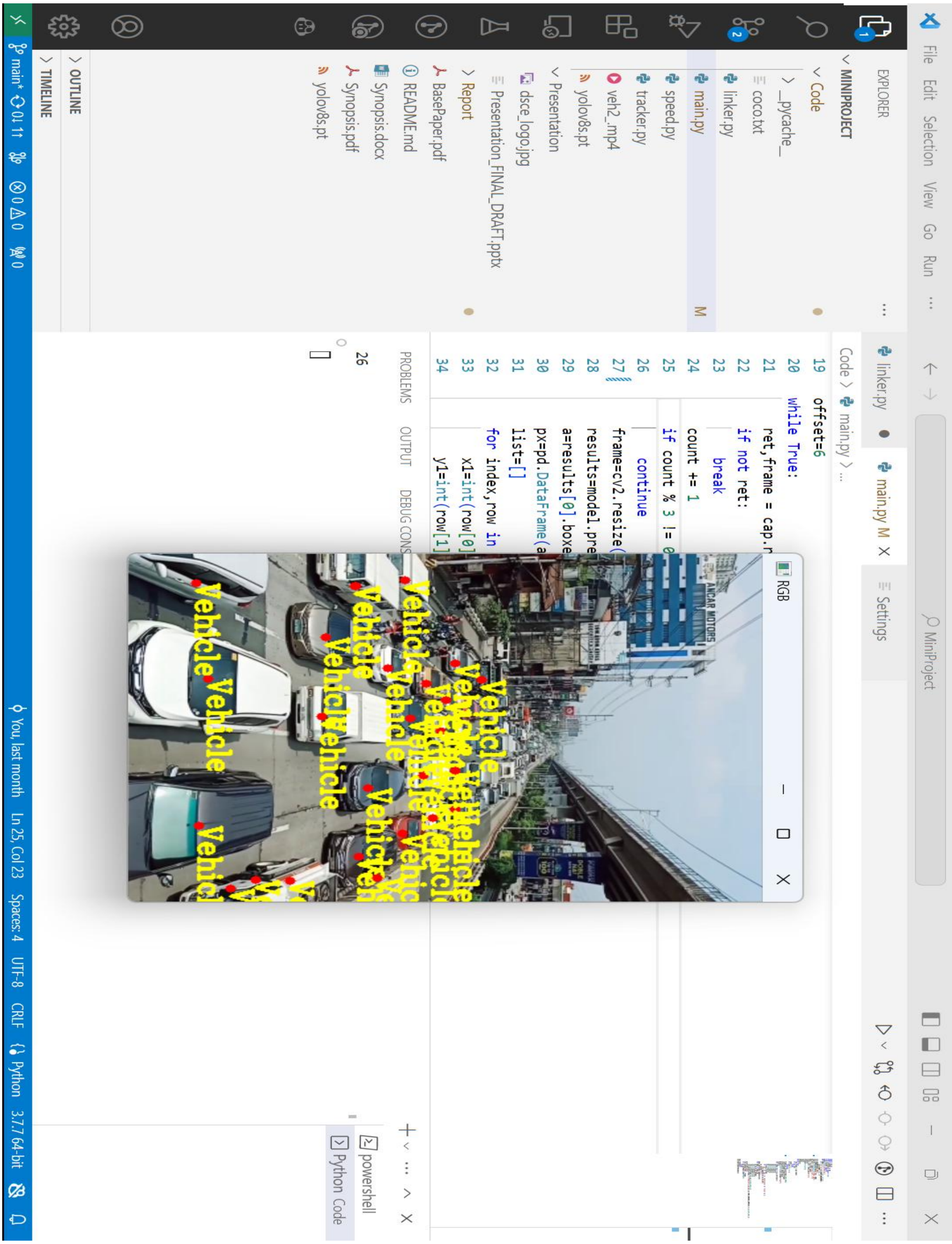
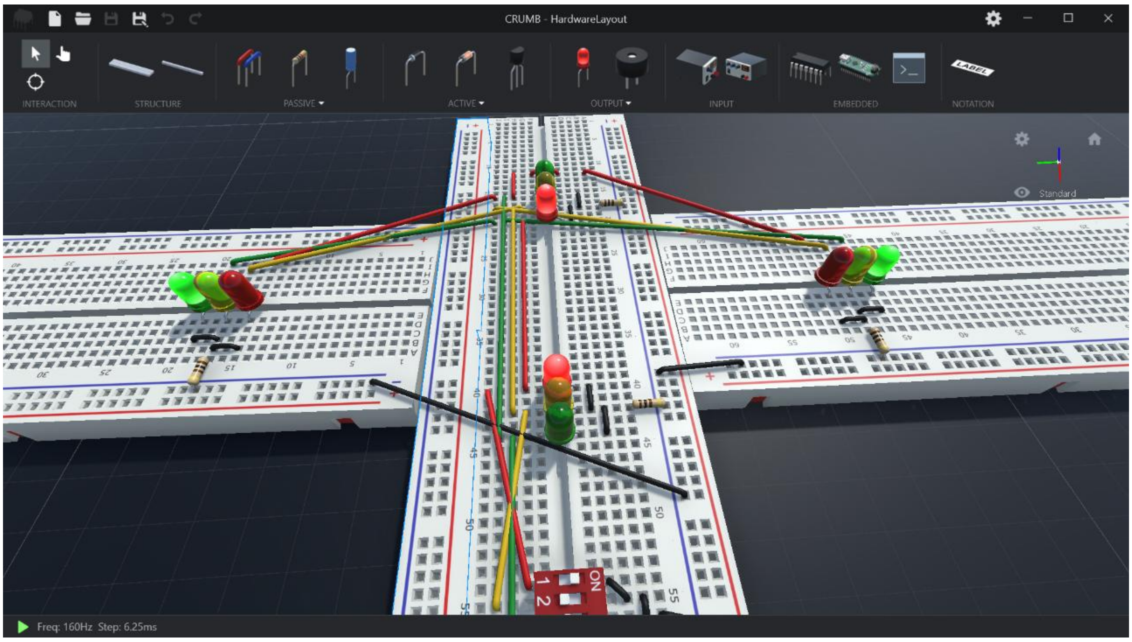


Fig (a). Detection model predicting density




- **Initial Implementation Costs** Implementing a smart traffic management system using computer vision can be expensive, involving the cost of cameras, computer hardware, software development, and installation. These costs may pose challenges, especially for cash-strapped municipalities or regions with limited budgets.

- **Maintenance** Once deployed, the system will require regular maintenance to ensure cameras are functioning correctly, software is up to date, and hardware remains operational. Maintenance costs, both in terms of time and money, should be factored into the project.

- **Privacy Concerns** The use of CCTV cameras for traffic monitoring raises privacy concerns. It's essential to address privacy issues and ensure that the system complies with privacy regulations. Proper data handling techniques may be necessary to protect individual privacy.

- **Data Security** Handling and transmitting traffic data over networks introduce potential security risks. Unauthorized access to the system or data breaches could compromise sensitive information, such as live camera feeds.

- **Limited Effectiveness During Extreme Conditions** The system may struggle to perform optimally during severe weather conditions (e.g., heavy rain, snow, fog) or in situations where visibility is significantly reduced.

- **Power Supply Dependence** The system relies on a stable power supply. Power outages or fluctuations could disrupt its operation. Implementing backup power solutions may be necessary to address this limitation.

- **Scalability** Expanding the system to cover a larger area with multiple junctions might be complex. Ensuring scalability without introducing inefficiencies or complications can be challenging.

Figure 3.1: a

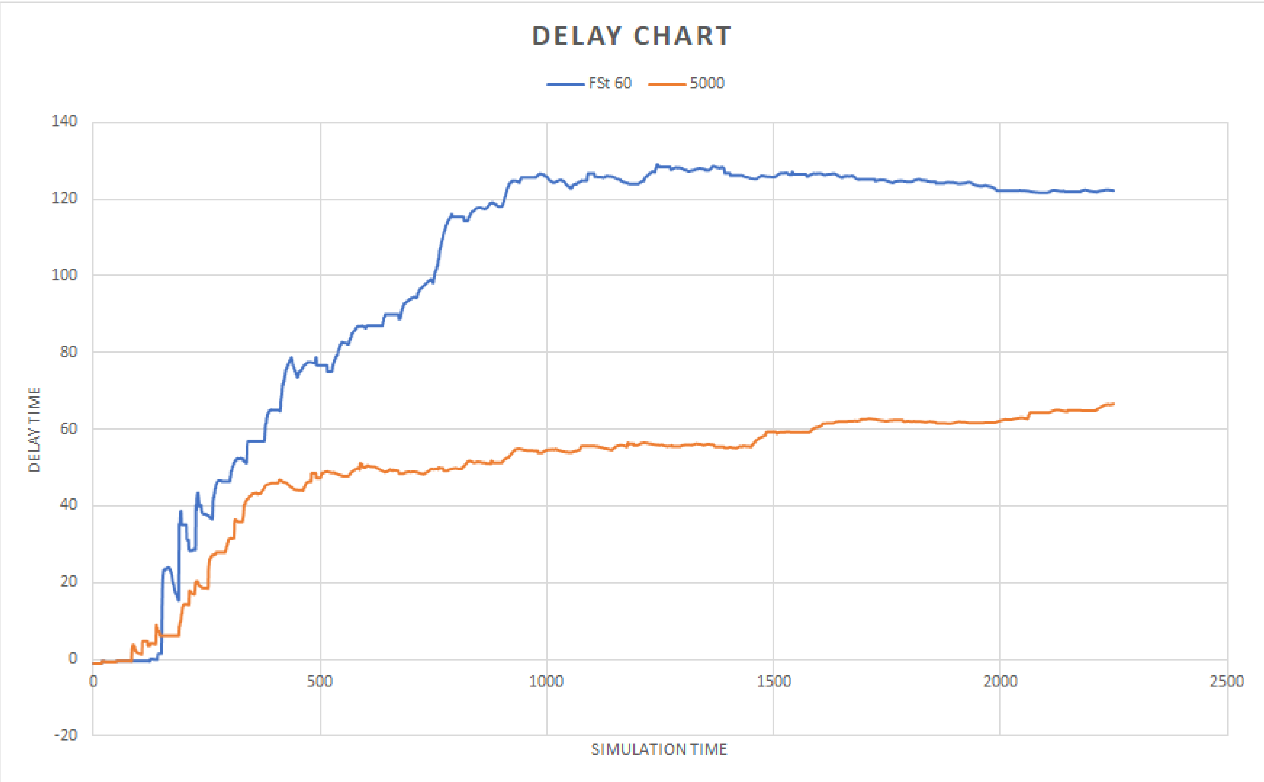
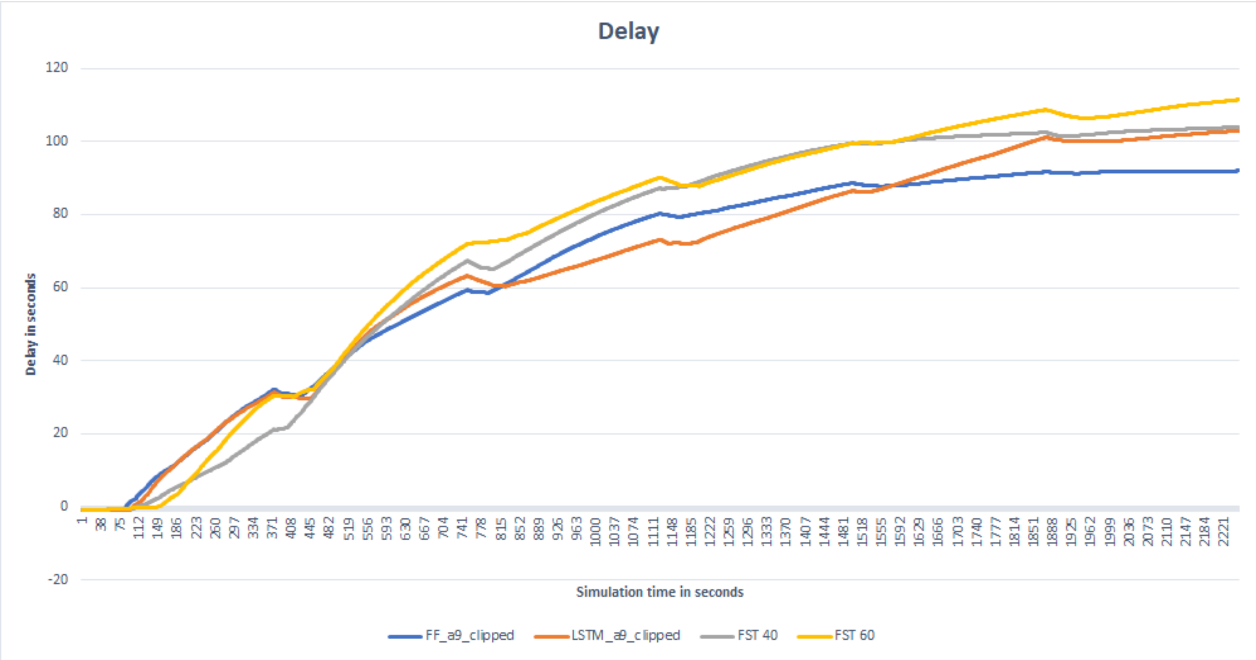
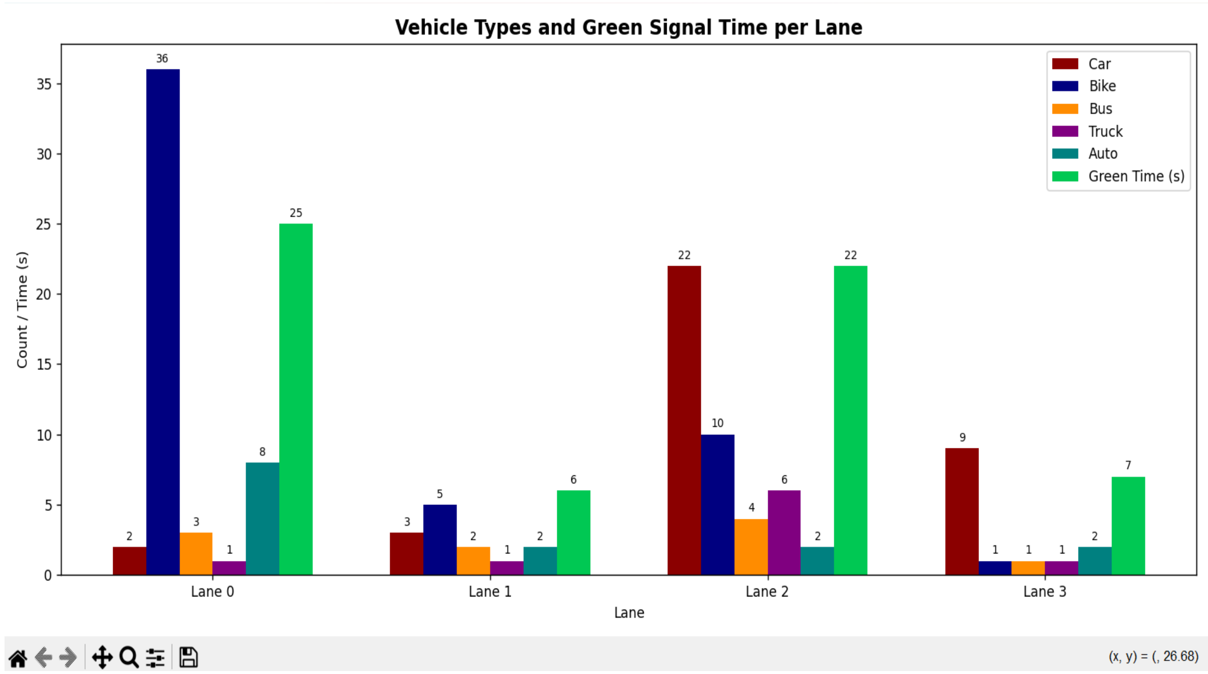


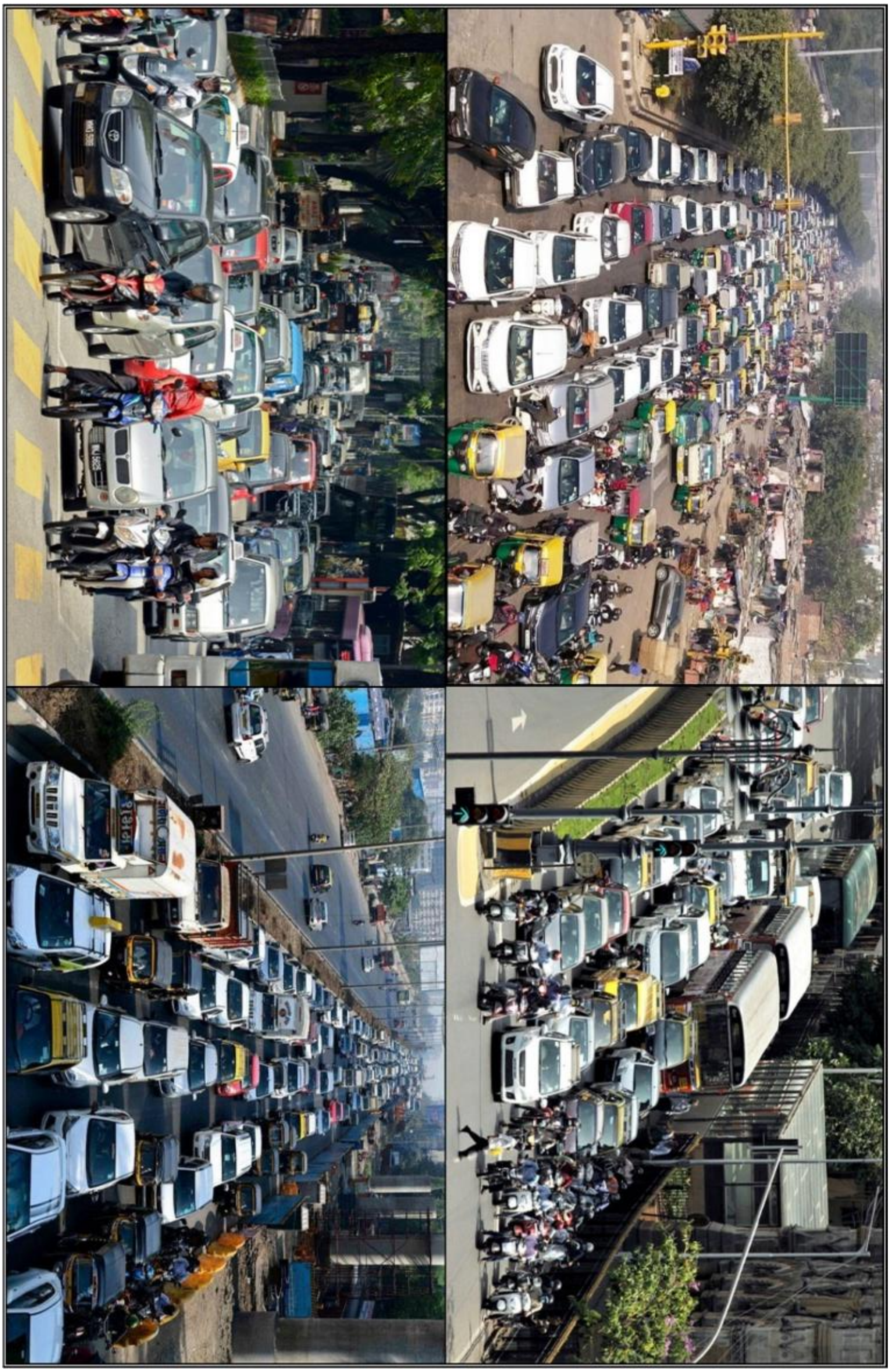
Figure 3.1: a




a) The graphs represents the delay time against the simulation time for 60 minutes in simulator taken at a junction when FST 60 is used in comparison to the RL agent which gives much lesser delay time.

b) The graph represents the delay time against the simulation time for 60 minutes in simulator





# CHAPTER 4

## CONCLUSION AND LEARNING OUTCOME

Our focus is on designing a practical and robust adaptive traffic signal control system specifically tailored to the conditions and requirements of the developing countries. Project indeed aims on creating a smart traffic signal and then actually put the research work into production.

## 4.1 FUTURE ENHANCEMENTS

While the current implementation of the traffic management system has the ability to optimize traffic flow, there are several further improvement and expansion

1. Testing on Raspberry Pi: Extending the practical application of the traffic management system, it is essential to test and optimize its performance on hardware like Raspberry Pi.

2. Improved Night time Accuracy with Thermal/IR Cameras: Enhancing the system's performance during night time or low-visibility conditions is critical. Integrating thermal or infrared (IR) cameras can provide better recognition of vehicles and pedestrians in the dark.

3. Data Encryption for Network Security: As data communication is integral to the system's functionality, implementing strong data encryption protocols is vital. Ensure that all data transmitted over the network is encrypted.

4. Port Scanning for Network Discovery: To enhance the system's network capabilities, consider implementing port scanning functionality. This feature allows the system to actively discover and identify available ports on the network.

5. Expansion to Multiple Junctions: Scaling the system's deployment to cover additional junctions and intersections is a logical step for urban traffic management. Expanding the system's coverage to multiple junctions (5, 6, or more)

6. Mutual Exclusion: To enhance the system's robustness and reliability, consider implementing mutual exclusion with a tool like Semaphore. This would ensure that there is no conflict during writing files on network.

## 4.2 LEARNING OUTCOMES:

- Environmental Awareness: Recognize the role traffic management plays in reducing emissions and promoting sustainable transportation practices.
- Evaluation and Analysis: Develop skills to evaluate the effectiveness of traffic light strategies by analyzing data and making evidence-based decisions.
- Traffic Management Skill: Develop a strong understanding of traffic management principles, including signal timing, intersection design, and traffic flow dynamics.
- Future Relevance: Gain knowledge and skills that are applicable in a rapidly urbanizing world where traffic management solutions are of increasing importance.
- Using Tools Efficiently: Tool like Copilot for improving coding skill, ChatGPT for proof reading and generating certain parts of this report, Google Bard and Mermaid.js to create images.
- Overall, participation in this project provides a well-rounded learning experience that encompasses technical skills, problem-solving abilities, teamwork, and a broader understanding of urban transportation challenges and solutions.
- Engaging in a project focused on solutions for traffic congestion using traffic lights can result in several valuable learning outcomes.

**4.3 CONCLUSION:**

The proposed system sets the green signal time adaptively according to the traffic density at the signal and ensures that the direction with more traffic is allotted a green signal for a longer duration of time as compared to the direction with lesser traffic. This will lower the unwanted delays, and delays, and reduce congestion and waiting time which in turn will reduce the fuel consumption and pollution.

The new system is expected to show much improvement over the current system in terms of the number of vehicles crossing the intersection, which is a significant improvement. This system can thus be integrated with the CCTV cameras in major cities in order to facilitate better management of traffic.

The solutions explored in the project span a wide spectrum, from short-term interventions like optimizing traffic signals timings to long-term strategies like promoting sustainable transportation modes and urban planning revisions.

The project has significant impacts on traffic congestion, ranging from increased travel times and decreased productivity to heightened pollution levels and compromised public safety.

Vehicle density in the cities, especially in the developing world, is increasing exponentially. This is both due to the population growth accompanied by increasing urbanization, as well as rise in the car ownership with improving economic prosperity.

However, the road infrastructure is often not able to keep pace with this increase due to financial as well as space constraints. In such a situation adaptive traffic signal control becomes the key to achieving maximum utilization of the existing road infrastructure.

A traffic controller is a system designed to regulate traffic at a single or a set of traffic intersections using static or adaptive rules. While the former creates a fixed phase switching routine based upon the historical traffic data, the adaptive controllers measure the traffic density and vehicle delays at each approach, and then generate an optimal green time for various phases so as to maximize the vehicle throughput at the junction.

The project aims to provide the techniques to minimize traffic congestion on Indian traffic scenarios and is being tested on the data collected in the city of Bangalore; using the Reinforcement Learning.

# References


1. Redmon J, Farhadi A (2018) Yolov3: an incremental improvement. arXiv preprint arXiv:1804.02767
2. Rath M (2018) Smart traffic management system for traffic control using automated mechanical and electronic devices. IOP Conf Ser Mater Sci Eng 377:012201. https://doi.org/10.1088/1757-899X/377/ 1/012201
3. Shinde MS, Jagtap S (2016) Intelligent traffic management systems. A review. Int J Innov Res Sci Technol 2:293–298
4. Komsiyah S, Desvania E (2021) Traffic lights analysis and simulation using fuzzy inference system of mamdani on three-signaled intersections. Procedia Comput Sci 179:268–280. https://doi.org/10.1016/ j.procs.2021.01.006
5. Toh CK, Sanguesa JA, Cano JC, Martinez FJ (2020) Advances in smart roads for future smart cities. Proc R Soc A 476(2233):20190439. https://doi.org/10.1098/rspa.2019.0439
6. Allström A, Barceló J, Ekström J, Grumert E, Gundlegård D, Rydergren C (2017) Traffic management for smart cities. In: Angelakis V, Tragos E, Pöhls HC, Kapovits A, Bassi A (eds) Designing, developing, and facilitating smart cities. Springer, Cham, pp 211–240. https://doi.org/10.1007/978-3-319-44924- 1_11
7. Asha CS, Narasimhadhan AV (2018) Vehicle counting for traffic management system using YOLO and correlation filter. In: 2018 IEEE international conference on electronics, computing and communica- tion technologies (CONECCT). IEEE, Bangalore, pp 1–6. https://doi.org/10.1109/CONECCT.2018. 8482380
8. Alpatov BA, Babayan PV, Ershov MD (2018) Vehicle detection and counting system for real-time traffic surveillance. In: 2018 7th Mediterranean conference on embedded computing (MECO). IEEE, Budva, pp 1–4. https://doi.org/10.1109/MECO.2018.8406017
9. Basil E, Sawant S (2017) IoT based traffic light control system using Raspberry Pi. In: 2017 Inter- national conference on energy, communication, data analytics and soft computing (ICECDS). IEEE, Chennai, pp 1078–1081. https://doi.org/10.1109/ICECDS.2017.8389604. https://ieeexplore.ieee.org/ document/8389604/
10. Corovic A, Ilic V, Duric S, Marijan M, Pavkovic B (2018) The real-time detection of traffic participants using YOLO algorithm. In: 2018 26th Telecommunications forum (TELFOR). IEEE, Belgrade, pp 1–4. https://doi.org/10.1109/TELFOR.2018.8611986
11. Makaba T, Doorsamy W, Paul BS (2021) Bayesian network-based framework for cost-implication assessment of road traffic collisions. Int J ITS Res 19(1):240–253. https://doi.org/10.1007/s13177- 020-00242-1
12. Khan M, Ectors W, Bellemans T, Janssens D, Wets G (2018) Unmanned aerial vehicle-based traf- fic analysis: a case study for shockwave identification and flow parameters estimation at signalized intersections. Remote Sens 10(3):458. https://doi.org/10.3390/rs10030458
13. Takano T, Ono S, Kawasaki H, Ikeuchi K (2021) High-resolution image data collection scheme for road sensing using wide-angle cameras on general-use vehicle criteria to include/exclude collected images for super resolution. Int J ITS Res 19(2):299–311. https://doi.org/10.1007/s13177-020-00243-0
14. Fielbaum A (2020) Strategic public transport design using autonomous vehicles and other new tech- nologies. Int J ITS Res 18(2):183–191. https://doi.org/10.1007/s13177-019-00190-5
15. Lykov S, Asakura Y (2020) Anomalous traffic pattern detection in large urban areas: tensor-based approach with continuum modeling of traffic flow. Int J ITS Res 18(1):13–21. https://doi.org/10.1007/ s13177-018-0167-5
16. Contreras M, Gamess E (2020) An algorithm based on VANET technology to count vehicles stopped at a traffic light. Int J ITS Res 18(1):122–139. https://doi.org/10.1007/s13177-019-00184-3
17. Hu H, Gao Z, Sheng Y, Zhang C, Zheng R (2019) Traffic density recognition based on image global texture feature. Int J ITS Res 17(3):171–180. https://doi.org/10.1007/s13177-019-00187-0
18. Peque G, Miyagi T, Kurauchi F (2018) Adaptive learning algorithms for simulation-based dynamic traffic user equilibrium. Int J ITS Res 16(3):215–226. https://doi.org/10.1007/s13177-017-0150-6

19. Hamidi H, Kamankesh A (2018) An approach to intelligent traffic management system using a multi- agent system. Int J ITS Res 16(2):112–124. https://doi.org/10.1007/s13177-017-0142-6
20. García-Nieto J, Alba E, Carolina Olivera A (2012) Swarm intelligence for traffic light scheduling: appli- cation to real urban areas. Eng Appl Artif Intell 25(2):274–283. https://doi.org/10.1016/j.engappai. 2011.04.011
21. Jain P (2011) Automatic traffic signal controller for roads by exploiting fuzzy logic. In: Das VV, Stephen J, Chaba Y (eds) Computer networks and information technologies, vol 142. Springer. Berlin, Heidelberg, pp 273–277. https://doi.org/10.1007/978-3-642-19542-6_46
22. Hilmani A, Maizate A, Hassouni L (2020) Automated real-time intelligent traffic control system for smart cities using wireless sensor networks. Wirel Commun Mob Comput 2020:1–28. https://doi.org/ 10.1155/2020/8841893
23. Singh S, Vishwakarma S, Kispotta S, Yadav A (2020) A statistical analysis of various technologies to detect and prevent fake news. In: Batra U, Roy NR, Panda B (eds) Data science and analytics, vol 1230. Springer, Singapore, pp 164–172. https://doi.org/10.1007/978-981-15-5830-6_15
24. Pop MD (2018) Traffic lights management using optimization tool. Procedia Soc Behav Sci 238:323–330. https://doi.org/10.1016/j.sbspro.2018.04.008
25. Alexy AB (2021) Github repository for YOLO V3. https://github.com/AlexeyAB/darknet
26. PrashantJha SS, Kispotta S (2019) A lightweight approach for an adaptive traffic signal. Jabalpur, India
27. Liu X, Wang H, Dong C (2021) An improved method of nonmotorized traffic tracking and classification to acquire traffic parameters at intersections. Int J ITS Res 19(2):312–323. https://doi.org/10.1007/ s13177-020-00247-w
28. Wang Z, Cui J, Zha H, Kagesawa M, Ono S, Ikeuchi K (2014) Foreground object detection by motion- based grouping of object parts. Int J ITS Res 12(2):70–82. https://doi.org/10.1007/s13177-013-0074- 8

# Narender Bommera

## PRJT.pdf

My Files

My Files

Kakatiya Institute of Technology and Science

---

### Document Details

Submission ID

**trn:oid:::3618:91356727**

Submission Date

**Apr 15, 2025, 4:14 PM GMT+5:30**

Download Date

**Apr 15, 2025, 4:16 PM GMT+5:30**

File Name

**PRJT.pdf**

File Size

**611.7 KB**

**23 Pages**

**4,775 Words**

**27,630 Characters**

# 11% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

## Match Groups

**36** Not Cited or Quoted 9%
Matches with neither in-text citation nor quotation marks

**0** Missing Quotations 0%
Matches that are still very similar to source material

**12** Missing Citation 2%
Matches that have quotation marks, but no in-text citation

**0** Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

## Top Sources

5% Internet sources
4% Publications
9% Submitted works (Student Papers)

## Integrity Flags

**0 Integrity Flags for Review**

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

## Match Groups

- **36** Not Cited or Quoted 9%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **12** Missing Citation 2%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

## Top Sources

| | |
|---|---|
| 5% | Internet sources |
| 4% | Publications |
| 9% | Submitted works (Student Papers) |

## Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

| | | | |
|---|---|---|---|
| 1 | Publication | Mihir M. Gandhi, Devansh S. Solanki, Rutwij S. Daptardar, Nirmala Shinde Baloork... | 2% |
| 2 | Submitted works | Colorado Technical University Online on 2025-04-12 | 2% |
| 3 | Submitted works | Canterbury Christ Church University on 2023-09-07 | 1% |
| 4 | Internet | ceur-ws.org | <1% |
| 5 | Submitted works | Chulalongkorn University on 2024-12-06 | <1% |
| 6 | Publication | Rama Devi Burri, Satish Reddy Nalamalapu, Musham Prashanthi, Bussa Sathwik. ... | <1% |
| 7 | Submitted works | University of Greenwich on 2024-01-22 | <1% |
| 8 | Submitted works | Vidyalankar Institute of Technology on 2025-04-12 | <1% |
| 9 | Submitted works | Sim University on 2024-10-13 | <1% |
| 10 | Submitted works | University of Hertfordshire on 2024-01-29 | <1% |

| | | |
|---|---|---|
| 11 | Submitted works | |
| University of Greenwich on 2024-01-22 | | <1% |
| 12 | Internet | |
| roboticulized.com | | <1% |
| 13 | Submitted works | |
| University of Greenwich on 2024-01-22 | | <1% |
| 14 | Internet | |
| delong.typepad.com | | <1% |
| 15 | Internet | |
| dev.ifet.gr | | <1% |
| 16 | Internet | |
| www.coursehero.com | | <1% |
| 17 | Submitted works | |
| Liverpool John Moores University on 2024-03-06 | | <1% |
| 18 | Submitted works | |
| Swiss School of Business and Management - SSBM on 2025-01-09 | | <1% |
| 19 | Submitted works | |
| UC, Boulder on 2024-05-06 | | <1% |
| 20 | Submitted works | |
| Universiti Teknologi MARA on 2017-07-11 | | <1% |
| 21 | Submitted works | |
| University of Greenwich on 2024-01-21 | | <1% |
| 22 | Submitted works | |
| University of Greenwich on 2024-01-22 | | <1% |
| 23 | Submitted works | |
| University of Hertfordshire on 2024-04-08 | | <1% |
| 24 | Submitted works | |
| Glasgow Caledonian University on 2019-11-18 | | <1% |

| | | |
|---|---|---|
| 25 | Submitted works | |
| University of Greenwich on 2024-01-22 | | <1% |