Dynamic Traffic Light Control: A Novel Approach for Congestion Mitigation and Traffic Optimization

Aadithya Hari Nair
Computer Science and Engineering
Amal Jyothi College of Engineering
, India
2024@cs.ajce.in

Adithi R Kumar
Computer Science and Engineering
Amal Jyothi College of Engineering
India
adithirkumar@gmail.com

Aleena Thomas
Computer Science and Engineering
Amal Jyothi College Of Engineering
India
aleenathomas@gmail.com

Jeffy Shiju
Computer Science and Engineering
Amal Jyothi College of Engineering
, India
Jeffyshiju1@gmail.com

Tom Kurian
Assistant Professor, Computer Science and Engineering
Amal Jyothi College of Engineering
India
tomkurian@amaljyothi.ac.in

Abstract—In cities, traffic congestion is a major problem that negatively affects both the quality of life and the effectiveness of transportation. To tackle this difficulty, we introduce a unique method for dynamic traffic signal regulation that takes waiting time and vehicle density into account. Our approach makes use of cutting-edge technology including Raspberry Pi for hardware integration, YOLOv8 and CUDA for vehicle detection, and SUMO for traffic simulation. Using SUMO, we first create a 2D map of a traffic scenario as input for further analysis. Next, we utilize YOLOv8 and CUDA to identify automobiles inside the simulated setting, allowing us to compute vehicle density by taking into account the number of vehicles and the length of the road. Congestion status is evaluated using a predetermined threshold value that was established by trial and error. To efficiently regulate traffic flow, traffic lights are dynamically turned green when the density is over the threshold, signaling congestion. However, there is a 30-second waiting period. Our algorithm's output is smoothly connected with hardware through the use of a Raspberry Pi, which manages traffic light modules to instantly make modifications. This integration makes it easier to regulate traffic signals effectively, which guarantees optimal traffic flow and reduces congestion.Our test findings show how well our method works in different lanes to optimize traffic flow and lessen congestion. Our technology presents a prospective option for enhancing urban traffic management and transportation efficiency by dynamically modifying traffic lights based on real-time data.

Keywords–CUDA,SUMO

I. Introduction

In the Urban environment, effective traffic management is very crucial to ensure a smooth flow of vehicles thereby reducing congestion. Traditional traffic signal systems work on fixed timings therefore causing congestion during peak hours. An innovative method to reduce this congestion by using vehicle density and waiting time as parameters is discussed in this paper. This approach aims to adapt signal timings according to real-time traffic conditions. During night time when vehicle congestion rates are usually lower at that time, drivers experience unwanted waiting time for the traffic signal to turn green. This inefficiency increases fuel consumption and leads to unnecessary environmental pollution. This drawback is due to the lack of a dynamic real-time traffic signal control system. An innovative method for real-time traffic signal control using vehicle density and waiting time is analyzed.

Through data analytics, our system monitors traffic conditions and makes intelligent decisions to regulate traffic flow. This reduces unnecessary delays and improves air quality and fuel efficiency. This proactive approach contributes to overall improvements in urban traffic management and sustainability.

II. SURVEY OF LITERATURE

[1] Pablo Barbecho Bautista et al. proposed an intelligent trans- portation system (ITS) [1] where vehicular communication and traffic simulation are used to evaluate the vehicular network. They proposed realistic models with open-source access.

Khalid Mohammed Almatar proposed an idea where real- time traffic data was gathered from sensors, cameras, and GPS devices, followed by rigorous cleaning and feature normaliza- tion [2]. Matrices like precision-recall and F1 score were used to evaluate model effectiveness, with the best algorithm to be applied in a real-time traffic system.

Mahmuda Akhtar et al. discussed the need to choose ap- propriate ML models (SML or DML) and train the selected model, assess the model performance, and use the model for real-time predictions By continuous monitoring the model is kept updated for making decisions to support traffic manage- ment it is compared with the other models [3].

Akbar Ali et al. discussed a method to collect real-time traf- fic data using MCS-DTEF technology [4], including vehicle density and speed, and creating traffic models for both NTC'S and MCS-DTEF scenarios. They utilized a SUMO simulator for traffic simulation at different times like morning, afternoon, and evening to compare results, starting from time, destination, fuel consumption, and average vehicle speed to evaluate the impact of MCS-DTEF on traffic efficiency.

Christopher Winfrey et al. Proposed a methodology where vehicle prob data was obtained from Regional Integrated Transportation (RITIS), sourced from INRIX, covering ap- proximately 15,000 road segments in Tennessee [5]. The signal extraction involved parsing files containing prob vehicle readings and TMC segments and is verified manually using the RITS Trend Map Tool. All the analyses are implemented by Python libraries such as numpy, pandas, and mathplotlib.

Prakash Rosayyan et al. suggested an edge computing-based emergency vehicle priority system [6]. It has experimented with an optimal controlled strategy proposed which reduces the waiting time of normal traffic along with Emergency Vehicle preemption using edge computing. IoT sensors and experiments were conducted on a real road with a dashboard unit and edge server. Priority to emergency vehicles is given without affecting the normal traffic

III. METHODOLOGY

A. Software Implementation

SUMO (Simulation of Urban Mobility) is an open-source traffic simulation tool widely used for modeling and simulating traffic flow road networks and transportation systems. SUMO provides a greater environment for traffic scenarios by which it can be modified, modeled, and analyzed in real-time.

SUMO provides real-time video playback capability where we can adjust simulation parameters and visualize the result spontaneously.

Using sumo we create a 2D traffic scenario of vehicles. Using this we can stimulate traffic vehicle flow, movements, and environment within a given urban area. We add this scenario as an input file by recording it and then exporting it.

This input file is given as input to the Python object detection code. It analyzes the recorded traffic scenario and provides the detection of vehicles in the provided environment. The Python code detects the vehicle and provides the code using YOLO (You Only Look Once) for object detection.

Using OpenCV's video capture the code captures video frames from an input which is a video file. The captured frame is resized to a specified width and height. The resizing is performed to optimize processing speed. YOLOv8 is used for vehicle detection in each resized frame. The output obtained is the total count of each vehicle from the video input. The number of vehicles in a frame per second(FPS) is displayed using the imshow method. As the final output total vehicle count will be obtained.

After detecting the vehicle number, the density of that area is determined. Density is calculated by dividing the vehicle count within the simulated area

B. Density Calculation

Density is the measure of number of vehicles in a particular stretch of road. Here we use the parameter (*D*), Density, and integrate it with SUMO (Simulation of Urban Mobility) which is a major aspect of dynamic traffic management. He proposed realistic models

Density Calculation Formula

The density of vehicles (*D*) is calculated using the following formula:

N

$$D = \frac{N}{L}$$

Where:

- D is the density of vehicles,
- N is the number of vehicles in the simulation,
- L is the length of the road segment detected by the camera.

Dynamic Traffic Light Control

The dynamic adjustment of traffic lights is triggered based on specific conditions related to the calculated density and waiting times.

1) **Waiting Time Threshold:** If the waiting time for vehi- cles at a particular junction exceeds 40 seconds.
2) **Density Threshold:** Also When the calculated density of vehicles exceeds a predefined threshold value, here we consider 0.22 as the threshold value already found using the trial and error method.

   Both the factors mentioned above activate the traffic light signal.

Implementation Logic

The algorithm follows these steps:

1) **Traffic Simulation Input:** Input data used here is based on SUMO simulation, providing real-time information on the number of vehicles (N) and the length of the road segment (L).
2) **Density Calculation:** The density of vehicles (D) is calculated using the formula mentioned earlier.
3) **Threshold Checking and Waiting time:** The system checks if the waiting time exceeds 40 seconds or if the calculated density surpasses the predefined threshold of 0.22.
4) **Dynamic Traffic Light Control:** If either threshold condition is met, the traffic light timings are dynamically adjusted to optimize the flow of traffic.

This approach will help change the traffic signal timings thereby reducing the congestion rate on roads thus creating a more realistic and dynamically changing system that con- tributes to the ever-changing urban mobility

C. Finding Threshold value

To find the threshold value of single-lane and double-lane roads by analyzing the change in density with the number of vehicles(using the trial and error method)

A threshold value below 0.11 is crucial for optimal traffic flow in a two-lane road.

No. of vehicles	Single Lane	Two Lane
1	0.04	0.02
3	0.12	0.06
6	0.24	0.12

TABLE I THRESHOLD VALUE CALCULATION

D. Hardware Implementation

1 By integrating the software with the hardware, the traffic signal module changes according to different densities and waiting time.

2 Raspberry Pi: Here Raspberry Pi is the central computing unit. and we Connect it to the power using C-type cables and ensure a stable network connection.

3 Breadboard and Numbered Wires: We create connections with the Raspberry Pi and traffic signal modules by establishing a proper connection with the breadboard.

4 Traffic Signal Modules: The traffic signal module is configured to work with GPIO pins on Raspberry Pi Each module corresponds to a specific traffic signal (red, green, yellow).

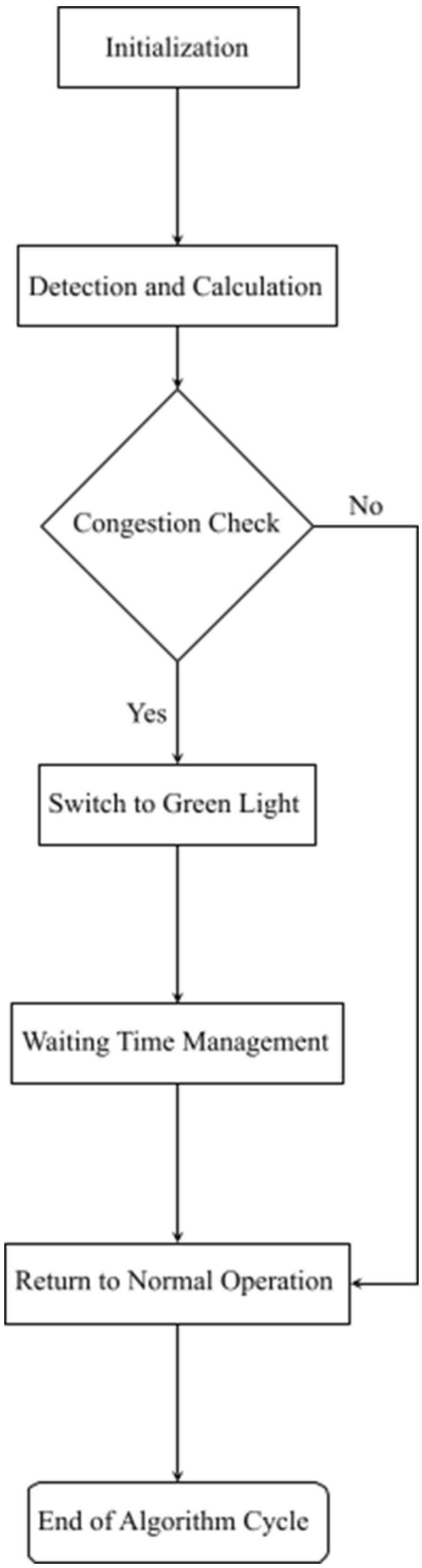
5 HDMI to HDMI Micro Cable:- HDMI to HDMI Micro cables is used for video output, and to integrate display modules thus showcasing real-time traffic information and signal status.

6 Breadboard: The breadboard enables a platform for organized circuit prototyping, allowing easy connectivity between various electronic components.

7 Traffic Signal Modules: We integrate four traffic signals to simulate and control traffic signal lights dynamically based on traffic conditions.

E. Dynamic Traffic Light Control Algorithm Flowchart

By using SUMO for traffic simulation and Python for object detection for vehicle counting we can analyze and quantify the traffic in an urban environment. This way we can evaluate the traffic congestion levels, and traffic dynamics thus providing dynamic management strategies.



IV RESULTS AND DISCUSSION

The findings of our study on dynamically adjusting traffic lights in real time depending on factors like vehicle count and density are presented in this section. We ran tests with a traffic simulation created by SUMO, then we used YOLOv8 and CUDA for vehicle recognition. Densities were computed using the collected vehicle counts, and the degree of congestion was ascertained by using a predetermined congestion threshold of 0.11. With a 30-second wait, traffic lights were dynamically changed to green to relieve congestion if density rose beyond the threshold.

A. Waiting Time vs. Density

Across four different lanes, the graph illustrates the rela- tionship between waiting time and traffic density. Because each lane is represented by a distinct colour, it is easy to distinguish between the various trends. Waiting time increases exponentially along the y-axis, representing the number of seconds that vehicles are delayed before moving forward via the traffic network, while density increases along the x-axis, which represents the number of vehicles per meter of road.

The blue-colored Lane 1 shows the strongest exponential increase in waiting time with increasing density. This implies that even slight increases in the number of vehicles on the road result in disproportionately greater wait periods, which could cause serious traffic jams and delays for commuters. After Lane 1, Lane 2 (red) shows an exponential curve that is marginally flatter, meaning that there is a somewhat less increase in waiting time for comparable density increases. The pattern, which emphasizes the cumulative impact of congestion on traffic flow and travel times, is still present, though.

The pattern is maintained by Lane 3 (green), where waiting times are growing more slowly than in Lanes 1 and 2, but they are still growing significantly as congestion increases. This implies that even in situations where congestion may be lessened, it still has a significant impact on traffic dynamics. Lastly, compared to the other lanes, Lane 4 (orange) has the least severe exponential growth in waiting time, suggesting a lesser susceptibility to changes in density. But even in this case, waiting times still rise with increasing density.

B. Comparison of Traffic Control Strategies: Traditional System vs. Proposed System

The relationship between waiting time (measured in seconds) and traffic density (measured in cars per meter) for two distinct traffic signal control systems is shown in the graph. The waiting time under a static traffic light timing scheme, where the waiting time is constant and independent of variations in traffic density, is represented by the blue line. On the other hand, under a dynamic traffic light control system, which modifies signal timing in real-time based on existing traffic circumstances, the red line denotes the waiting time. In order to maximize traffic flow and reduce congestion, waiting times dynamically increase as traffic density rises, producing comparatively shorter wait times. This comparison demonstrates how dynamic traffic light control can cut total wait times and increase the effectiveness of transportation.

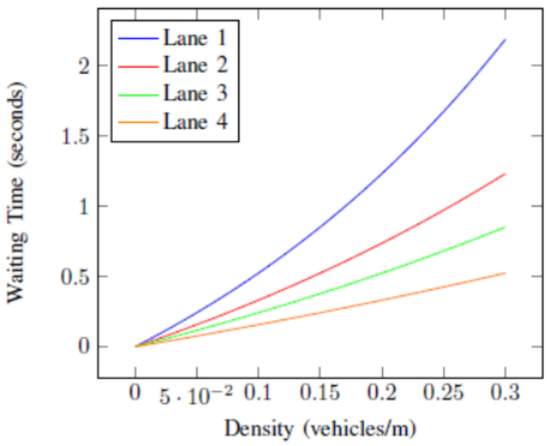
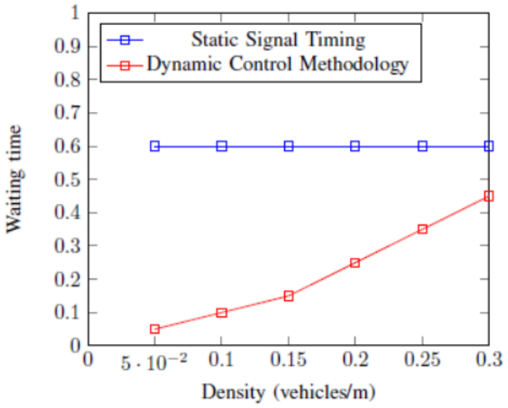


Fig. 2. Exponential Waiting Time vs Density for Different Lanes



REFERENCES

[1] How does the traffic behavior change by using SUMO traffic generation tools, Pablo Barbecho Bautista, Luis Urquiza Aguiar, Mo´nica Aguilar Igartua

[2] Traffic congestion patterns in the urban road network: (Dammam metropolitan area), Khalid Mohammed Almatar

[3] A Review of Traffic Congestion Prediction Using Artificial Intelligence, Mahmudad Akhtar, Sara Moridpour

[4] Pablo Barbecho Bautista, Luis Urquiza Aguiar, Mo´nica Aguilar Igartua

[5] Using big data and machine learning to rank traffic signals in Tennessee, Christopher Winfrey, Piro Meleby, Lei Miao

[6] An optimal control strategy for emergency vehicle priority system in smart cities using edge computing and IoT sensors, Prakash Rosayyan, Jasmine Paul, Senthilkumar Subramaniam, Saravana Ilango Ganesan

[7] Including congestion effects in urban road traffic CO2 emissions mod- elling: Do Local Government Authorities have the right options? Matt Grote, Ian Williams, John Preston, Simon