# Design and Implementation of an Unreal Engine 4-Based Smart Traffic Control System for Smart City Applications

Md. Imtiaz Hossain Subree[1], Md. Rakib Hasan[2], Maksuda Haider Sayma[3]
Dept. of ICT, Comilla University, Cumilla - 3506, Bangladesh

***Abstract*—Traffic congestion is a serious problem nowadays, especially in Dhaka city. With the increasing population and automation, it has become one of the most critical issues in our country. There can be a lot of causes of congestion in traffic, such as insufficient capacity, large red signal delay, unrestrained demand, etc, which causes extra time delay, extra fuel consumption, a speed reduction of vehicle, and financial loss. The traffic control system is one of the most important factors affecting traffic flow. Poor traffic management around these hotspots could result in prolonged traffic jams. Small critical locations that are frequent hotspots for congestion are a common byproduct of poorly constructed road networks in many developing countries. In this research, we first offer a straightforward automated image processing method for analyzing CCTV camera image feeds to determine the level of traffic congestion. Our system's design seeks to use real-time photos from the cams at traffic intersections to calculate traffic density using image processing, and to adjust the traffic signal on the data obtained based on the current traffic congestion on the road. We suggest tailoring our system to erratic traffic feeds with poor visual quality. Using live Surveillance camera feeds from multiple traffic signals in Dhaka city, we demonstrate evidence of this bottleneck breakdown tendency persisting over prolonged time frames across multiple locations. To partially address this problem, we offer a local adapting algorithm that coordinates signal timing behavior in a restricted area and can locally minimize congestion collapse by maintaining time-variant traffic surges. Using simulation-based research on basic network topologies, we show how our local decongestion protocol may boost the capacity of the road and avoid traffic collapse in limited scenarios.**

***Keywords*—*Traffic control system; traffic congestion; CCTV; image processing; simulation***

## I. Introduction

Within the cutting-edge generation, the exponential growth of traffic congestion is among the most vital problems facing our civilization today. Due to the rise in automobiles in urban regions, many road networks are experiencing issues with the decline in road capacity and the accompanying stage of the carrier.

Traffic congestion eats up 5 million working hours each day in Dhaka city, which costs the country 11.4B USD a year. Vehicular speed has decreased from 21kmph to 6.5kmph in the last 12 years, which causes 40% of fuel wastage. 40% of the fuel is being wasted. There are 70 signalized intersections in Dhaka city. In our country, generally, 2 types of systems are used for controlling traffic at intersections.

1) Manual Traffic Control System: In the manual traffic control system, to manage traffic, a certain region is assigned to traffic police. To regulate traffic, traffic police utilize signboards, whistles, lasers, and other tools.
2) Automatic Traffic Control System: In an automatic system, Traffic Lights are used to control the traffic. A timer is used to control the signals of the traffic lights. The lights get off and on based on the timer value. [1]

The conventional methods have distinct drawbacks. There is a significant labor requirement for the manual controlling system. We are unable to have traffic police manually regulate traffic in all sections of a town or city due to the inadequate number of traffic officers. Conventional traffic control employs a traffic signal with a timer for each phase that is static and cannot change based on the volume of traffic on that particular road in real time. It's always fixed and do not alter or update in real-time in accordance with traffic on the roads [1].

The main problem with the conventional automatic traffic control system is, the traffic lights use a timer for every segment, which is rigid and does update or change based on the real-time traffic on the road. During morning and evening, the majority of the traffic movement direction is always opposite. Taking into account this theory, one particular direction with heavier traffic is given the same amount of green signal time as the other direction, which has lesser traffic than the other or in some cases, no traffic at all, which is inconvenient.

So, the main problems identified by us were

- Manual Traffic control system is inconvenient and faulty.
- Automatic Traffic control system uses a fixed timer which is rigid and does not update with the real-time traffic on the road.

From this existing problems, the following research questions arise:

- Is there a better way to replace the existing system?
- Can we upgrade the existing automatic traffic control system with a timer that can adjust to the existing traffic condition and update based on that?

Our suggested method intends to provide a computer vision-based traffic signal controller that can adjust to the existing traffic condition. It makes use of actual footage from the Surveillance cameras installed at road intersections to determine the traffic density by measuring the number of cars at the light and altering the length of the green signal accordingly. The vehicles are separated into four categories: automobiles, trucks, buses, and bikes to calculate the correct calculation of the green light time. The system employs YOLOv5 to compute the number of automobiles before modifying the traffic signal's timer in line with the suitable direction of vehicle density. In contrast to a static system, traffic is cleared significantly more swiftly thanks to this, maximizing the periods whenever the green light is on. Unwanted delays, traffic, and waiting times are reduced as a result, which also reduces fuel consumption and emissions.

The main contribution of our proposed system are as follows:

- Proposing a better solution for timer based automatic traffic control system.
- Using the existing Surveillance cameras on the road to collect the traffic info and update the signals accordingly.
- Using YOLOv5, which is 88% smaller and 180% faster than YOLOv4, to compute the number of automobiles before modifying the traffic signal's timer in line with the suitable direction of vehicle density.
- Achieving almost perfect result by counting the number of vehicles, identifying the types of vehicles, measuring the speed of each vehicle and feeding proper value to the server.

## II. Literature Review

de Frías, C. J., Al-Kaff, A., Moreno, F. M., Madridano, Á., and Armingol, J. M. describe a cooperative system for monitoring traffic flow based on the use of unmanned aerial vehicles (UAVs) in [10]. A lightweight semantic neural network was suggested by this system, which uses RGB photos as input to produce a segmented image of the vehicles and their 2D actual positions. Then, each car receives this information so that it may fully understand its environment. The suggested approach has undergone multiple experiments with a variety of scenarios and conditions, and the outcomes demonstrate the system's effectiveness and robustness while also demonstrating how it may be used for traffic monitoring.

PJ Navarathna, VP Malagi proposed some important technology and answers to the many issues that the lack of digitalization was causing the populace in [9]. It addresses problems with the city's infrastructure, public security, and safety, and offers the best remedies. It places a strong emphasis on using AI in conjunction with other technologies, such as the Internet of Things (IoT), deep learning, machine learning, pattern recognition, big data analytics, and cloud infrastructures, to create a fully functional smart city.

Karimi, A., and Duggirala, P. S. presented a new approach to formalizing and implementing traffic rules in [12]. The DMV driver manual for California is used as a working example. Its method offers a clear mapping between the handbook's rules and their formal model, as well as between the model and implementation. They use the Clingo programming language to explicitly model the traffic regulations in the Answer Set Programming (ASP) logic programming paradigm to show the effectiveness of this method. They later incorporate these regulations into CARLA, a computerized testing ground for autonomous automobiles. By correctly applying the right-of-way rules for autonomous cars in real-time, they replicate the behavior of autonomous vehicles at four-way and three-way uncontrolled junctions. As a result, as compared to CARLA's default FIFO controller, the behaviors of autonomous vehicles under our controller are more realistic.

Manjón Prado, Javier examined the escalating issue of traffic congestion, as well as various potential solutions, such as traffic simulation software and autonomous vehicles in [11]. The new choice for recreating a traffic scenario is Unreal Engine. Using a click-and-drag editor, a tool has been developed that enables users to build and alter road tracks and add autonomous car models to them. Different scenarios are used to evaluate the simulator while important data is collected. Finally, the results are discussed together with suggestions and directives for future improvement.

Khushi used video processing to fix the congestion problem in [1]. Before being delivered to the servers, the footage from the live stream is processed. There, a CPP-based algorithm is utilized to generate the outcome. The dynamic algorithm demonstrated a 35

Renjith Soman employed ANN and a fuzzy controller to suggest a smart traffic signal system in [2]. Images obtained from cameras placed at traffic locations are used by this system. Before further normalization, the image is initially transformed to a grayscale version. Following that, the segmented image is passed through an ANN to count the automobiles regardless of size, and the output is then used in a control scheme to set periods for the green and red lights using crisp output. Results had a 2

Chabchoub, A., Hamouda, A., Al-Ahmadi, S., and Cherif, A. designed a smart traffic light controller using fuzzy logic and image processing with MATLAB, to control movement in two ways, aided by a camera and auto sensors. in [3]. The Fuzzy logic has two inputs and six outputs designed, the console input is the number of cars on each road and the time of the assumed red, yellow and green signal according to the vehicles congestion. The simulation result is similar to the proposed control unit, as it deals with the lights simultaneously according to the number of cars in each branch of the road, which leads to the use of all the time to operate the stoplights.

A. A. Zaid, Y. Suhweil, and M. A. Yaman suggested a traffic signal that can be adjusted to the prevailing traffic conditions using fuzzy logic [4]. For the major and secondary driveways in this system, two fuzzy control systems with three inputs and one output are used. Simulated traffic conditions were improved for lower traffic density using VISSIM and MATLAB.

Al-Majhad, H. G., Bramantoro, A., Syamsuddin, I., Yunianta, A., Basori, A. H., Prabuwono, A. S., and Barukab, O. M. aims to benefit from the current Riyadh road infrastructure and apply the Internet of Things paradigm for detecting traffic congestion with Everything as a Service approach. in [5]. By using the services, the users are able to identify the exact location where congestion occurs and an alternate solution can be provided easily. To achieve this, Business Process Execution Language is embedded as a supporting framework layer. This approach clearly defines how the services are executed through the proposed models.

Agrawal, Aditi, and Rajeev Paulus proposes Fuzzy Adaptive Control System (FACS) that uses fuzzy logic to decide the phase sequence and green-time for each lane based on sensed input parameters in [6]. It is designed with an aim to improve traffic clearance at isolated intersection especially in peak traffic hours of the day along with giving precedence to emergency vehicle as soon as it is detected and also assist pedestrian passage thus reducing their waiting time at the intersection. Performance of the proposed Fuzzy Adaptive Control System (FACS) is evaluated through simulations and compared with Pre-Timed Control System (PTCS) and Traffic Density-based Control System (TDCS) at a busy intersection with lanes leading to offices, schools and hospitals. Simulation results show significant improvement over PTCS and TDCS in terms of traffic clearance, immediate addressing of the emergency vehicle and giving preference to pedestrian passage at the intersection.

Adbeb, Tesfanesh, Di Wu, and Muhammad Ibrar presented heuristic algorithms called Congestion-Free Path (CFP) and Optimize CFP (OCFP), in SD-VANET architecture in [7]. The proposed algorithms address the traffic congestion issue and also provide a feasible path (less end-to-end delay) for a vehicle in VANET. They used the NS-3 simulator to evaluate the performance of the proposed algorithms, and for generating a real scenario of VANET traffic; we use the SUMO module. The results show that the proposed algorithms decrease road traffic congestion drastically compared to exiting approaches.

Saili Shinde examined the various traffic signal control system techniques in [8]. This study notes that each method has a similar architecture, which includes selecting input data, extracting traffic parameters from it, processing it, figuring out density, and updating parameters. The numerous traffic signal control system methods are examined in this reference. This paper shows that the architecture of each approach is similar and consists of picking data input, obtaining traffic data from it, analyzing it, calculating intensity, and changing parameters. The first method involves the employment of VANETS to collect information on each vessel's position, which is subsequently sent via installed GPS to the nearest Intelligent Traffic Light (ITL). Additionally, these ITLs will update the vehicles around with statistics. In the event of incidents, information would be provided to drivers so they could choose an alternative path to avoid traffic. This method is not workable because of the high deployment costs. The second technique uses a transceiver to capture each car's token utilizing infrared sensor-based embedded systems. Vehicles' data lines regularity tags can be used to identify them in an emergency and let vehicles pass. By adopting this technique, red light violations are discovered. However, this strategy is not flexible because infrared sensors need to be visible. The third methodology employs a fuzzy logic approach and two probabilistic reasoning controllers, one of which lengthens the green period of a route at an intersection and the other of which optimizes the signal. The outgoing and incoming lines' record cameras act as sensors to collect data from input sources. The controller then uses the data gathered by these detectors to determine the best course of action and minimize the goal function. The fourth method uses fuzzy reason, and the system's inputs include the number of vehicles and the average speed of movement in each route. Using sensors placed along the route, it is possible to calculate the number of cars and the cruising pace of the flow of traffic. The fifth approach makes use of spaced-apart photoelectric sensors to collect data and transmit it to the traffic panel, which weighs each road individually and adjusts the traffic light as necessary. However, maintenance is quite expensive. Video image processing is employed to detain the data in the sixth technique. Dynamic noise removal and numerous morphological processes are conducted to get a clear illustration of the automobile. A fresh rectangle is generated, and the vehicle calculation is increased each time a new car enters the subject of attention. Although the algorithm is simple to use, it does not account for shadow overlapping and occlusion.

Golubev, Kirill, Aleksandr Zagarskikh, and Andrey Karsakov described a brand-new framework for agent-based traffic modeling that combines various traffic model classes into a single vehicle agent and lets the user choose a particular model for each supported class in [13]. The framework was created with the help of the Unreal Engine 4 gaming engine, which enables users to simulate realistic vehicle interaction and create realistic visualizations for visual analysis and computational steering.

### *A. Limitations of Related Works*

- The constraints of a vehicular network scenario that new communication, data, and resource approaches must address to increase overall network performance, such as low power, full storage, message's time-to-live, and bandwidth.
- SD VANET has insufficient performance evaluation, Uncertainty over the proposal's usefulness and correctness, as well as integration problems with fog, SDN, and VANET.
- No security analysis, no solution when connection loss with controller occurs. only weak security analysis.
- ANN and a fuzzy controller has hardware dependency, Sharing Difficulty, Overfitting Likely, ANN Solution Not Guaranteed.
- Fuzzy logic has two significant drawbacks: it cannot handle ambiguous data, and it cannot infer human thought.

## III. Proposed System

Our proposed system by dynamically calculating the green signal period based on the volume of traffic at the signal hopes to reduce such instances. By doing this, it will be ensured that the way with greater traffic receives a longer green signal duration than the route with a relatively lower volume of traffic. The suggested solution uses image processing and object detection to calculate real-time traffic density from images captured by CCTV cameras placed at traffic intersections. Fig. 1 illustrates how this image is transmitted to the vehicle detecting algorithm that uses YOLOv5. To determine the traffic density, the number of each type of vehicle is counted, including automobiles, cyclists, buses, and trucks. This density, along with a few other variables, is used by the signal-switching algorithm to set the green signal timer for each lane. Accordingly, the red signal times are modified. To prevent a particular lane from going without traffic, there is a highest and lowest value for the green signal time.

So, the proposed system will work this way.

- The proposed system will use Closed Circuit Television (CCTV) for collecting live videos and snapshots installed at traffic signals.
- Snapshots from the CCTVs will be passed to servers at intersections for combining computer vision and image processing to calculate traffic density.
- For detecting vehicles, the YOLOv5 model will be used as shown before.
- A time scheduling algorithm will use these values and update the green and red signal times for each signal.

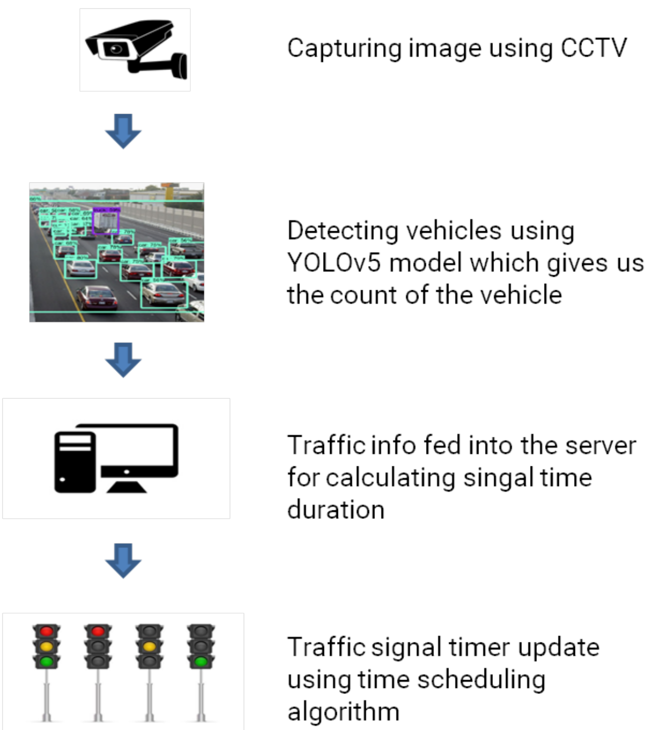


Fig. 1. Proposed system

Fig. 1 Shows the flow diagram of the proposed system.

### A. Vehicle Detection System

The proposed system uses YOLOv5 (You Only Look Once) for automobile recognition. This is one of the most well-known object detection algorithms for its accuracy and speed. It can perceive vehicles of different modules and can provide the desired accuracy and processing time. A single neural network is applied to the entire image by the algorithm, which then divides it into regions and forecasts enclosing frames and possibilities for each region. The projected probabilities are used to weigh these bounding boxes. By downloading photos from Google and manually labeling them using the visual image annotation tool LabelIMG, the dataset for training the replica was created. Then, the system was taught by pre-guided credence obtained from the YOLO site. By altering the 'classes' variable, the number of yield synapses in the ultimate coating was placed to be identical to the number of modules the system is intended to perceive. The replica was taught until the loss was much lower and didn't appear to be decreasing after making these configuration adjustments. The OpenCV library was then used to detect vehicles using these weights that were entered into the program. Following the model has been filled and a figure has been fed into it, it outputs the results in a JSON format, or as pairs of keys and values, where keys are labels and the values are the labels' confidence and coordinates.

Fig. 2 shows the test result of our vehicle detection system. Applying the YOLOv5 model, we can detect the different kinds of cars, in packages and with corresponding labels.

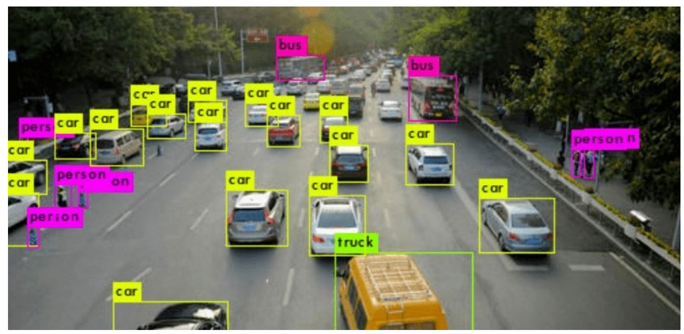


Fig. 2. Result of vehicle detection system

### B. Signal Updating

The servers at junctions will receive images from the Closed-circuit- circuit television and process them to calculate the traffic volume. The traffic surveillance module's dynamic traffic data is used by the Signal Shifting Technique to set the green signal schedule and update other lights' red signal timers. The method uses previously observed vehicle data that is provided in JSON format. Following that, the cumulative number of automobiles in each class is determined using this data. The program then identifies the lane with the highest car density. The data is subsequently transferred to the server after being located in the more crowded lane. The computer then extends the signal's green light by a few seconds where a high vehicular density was observed.

The default timing is established for the primary signal of the very first loop when the program is initially executed. Two separate lines are activated here, one to handle vehicle detection for each lane, and another to lever the timer of the existing signal. While the green signal timer is running for the current signal, the system takes a snapshot of the next lane and the red-light time for the other lights, and the green-light time for the next signal, respectively. As the timer and the vehicle detection are handled by two different lines, the timer does not interfere. When, the green light time of the existing signal becomes zero, the timer for the next signal starts. The cycle of counting timers and detecting vehicles keep going in the same manner for each signal.

The green light duration for each signal can be measured by this:

Here, Gdur is the duration of green signal time The number of vehicles is the total number of cars of each type detected by the vehicle detection system Average time to cross intersection = Distance⁄(Average speed of the vehicle) The number of lanes is the total number of tracks available in the intersection

The typical speed of vehicles will be measured in advance and will be stored on the server. Depending on the class of vehicles, the speed will be different. Using the vehicle detection system, we will detect the type of vehicles as well, and according to that data, we will use the average speed values to measure the average time to cross the intersection. The signals do not shift in the direction with the highest density initially; instead, they do so in cycles. This is in line with the existing structure, which sees the signals convert to green sequentially in a predictable fashion without requiring inhabitants to adjust their behavior or create any perplexity. The yellow signs have also been taken into account, and the signal sequence is the same as it is in the existing system.

After the signal turns green, the picture is snapped after some time. This provides the structure a period to decode the figure, compute the number of automobiles in every class present in the figure, determine the time for the green light, and then set the timers for both the existing signal and the next signal's red signal based on that calculation. The best green light duration based on the number of cars of each type at a signal was determined using the bandwidth of automobiles at starting and their sprint times. From that, an estimated of the typical time each type of vehicle requires to cross a junction was found.

$$G_{dur} = \frac{\sum_{vehicletype}(TotalVehicles * AverageCrossingTime)}{TotalLanes + 1} \quad (1)$$

Here,

$G_{dur}$ is the duration of the green signal time

TotalVehicles is the total number of cars of each type detected by the vehicle detection system

AverageCrossingTime = $\frac{Distance}{AverageVehicleSpeed}$

TotalLanes is the total number of tracks available in the intersection

The typical speed of vehicles will be measured in advance and will be stored on the server. Depending on the class of vehicles, the speed will be different. Using the vehicle detection system, we will detect the type of vehicles as well, and according to that data, we will use the average speed values to measure the average time to cross the intersection. The signals do not shift in the direction with the highest density initially; instead, they do so in cycles. This is in line with the existing structure, which sees the signals convert to green sequentially in a predictable fashion without requiring inhabitants to adjust their behavior or create any perplexity. The yellow signs have also been taken into account, and the signal sequence is the same as it is in the existing system.

After the signal turns green, the picture is snapped after some time. This provides the structure a period to decode the figure, compute the number of automobiles in every class present in the figure, determine the time for the green light, and then set the timers for both the existing signal and the next signal's red signal based on that calculation. The best green light duration based on the number of cars of each type at a signal was determined using the bandwidth of automobiles at starting and their sprint times. From that, an estimated of the typical time each type of vehicle requires to cross a junction was found.

### *C. Simulation*

To show the process we have created two different simulations using Unreal Engine 4 from scratch. One of which shows the current traffic system and the other shows our proposed system. It helps with system visualization and comparison with the current static system. Unreal Engine (UE) is a free-to-use 3D computer graphics game engine developed by Epic Games. The next-gen real-time 3D content and experiences are made possible by Unreal Engine, which is used by game developers and producers across industries. Unreal Engine is a full set of development tools for games, broadcast and live event production, education and simulation, architectural visualization and automotive, television content and linear film generation, and other real-time functions.

A four-way intersection with four traffic lights is present there. A timer that shows how much time is left until a signal turns between green-yellow, yellow-red, and red-green is situated on top of each signal. The quantity of vehicles that have passed through the intersection is also shown next to each light. Different types of vehicles arrive from all directions. To make things even, we had to keep the environment and the traffic flow similar for each simulation. And for simplicity, we have set the speed of each class of vehicles to the average speed of their class.

Factors to consider (same for both simulations): 1. Number of lanes 2. Traffic flow 3. Speed of vehicle 4. Lag suffered by vehicles during startup 5. Similar road, intersection, and the same distance

*1) 1st simulation (traditional system):* In the traditional system, every phase of the traffic signals uses a fixed timer that does not update or vary in response to the actual congestion on the road. So, one particular direction with heavier traffic is given the same amount of green signal time as the other direction, which has lesser traffic than the other or in some cases, no traffic at all, which is inconvenient.


Fig. 3. Traditional traffic system (1)

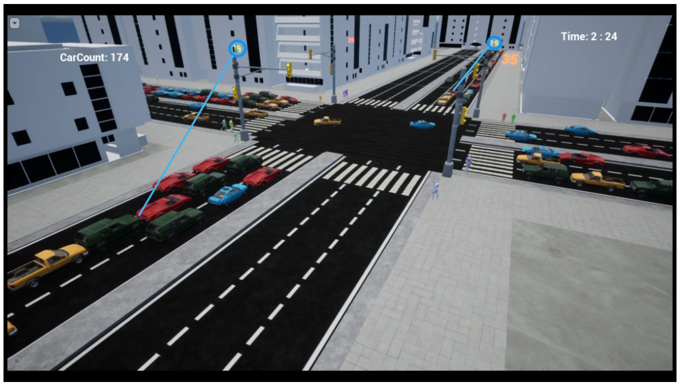


Fig. 4. Traditional traffic system (2)

Fig. 3 and Fig. 4 show the simulation of the traditional traffic system. In this simulation, we will see that, for every signal, the timer is fixed. Every signal is given the same amount of green signal time, in this case, 20 seconds each. But the flow of traffic in each direction is not the same, which doesn't allow proper traffic flow in the intersections as intended.

*2) 2nd simulation (proposed system):* By automatically calculating the green signal period based on the volume of traffic at the signal, the approach we've suggested tries to mitigate such issues. By doing so, it will be ensured that the route with greater traffic receives a longer green signal duration than the route with a relatively lower volume of traffic.

This simulation shows how our proposed system works. As the traffic flow of each lane is not the same, our proposed system updates the signal based on real-time traffic.

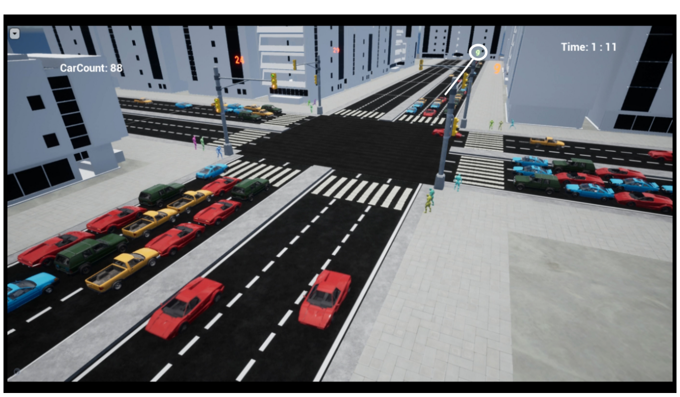


Fig. 5. Proposed traffic system (1)


Fig. 6. Proposed traffic system (2)

Fig. 5 shows the lane with a lower traffic flow is assigned a less amount of green signal time, 10 seconds in this case.

Fig. 6 shows the lane with a higher traffic flow is assigned a large amount of green signal time, 40 seconds in this case. This value can vary from 10 seconds minimum to 70 seconds maximum in our simulation based on the real-time traffic.

## IV. Result and Analysis

Using several sample photographs with various numbers of cars, our vehicle detection method was put to the test, and it was found that the detection performance ranged between 65 and 75 percent. Although sufficient, this is not ideal. The absence of a pertinent dataset is the primary cause of the lower accuracy. The system's accuracy can be improved by feeding the model data from actual traffic camera footage.

Two simulations were executed multiple times to assess the proposed method with the established system, and from those simulations, some data was acquired.

In both cases, we have took into account the factors to consider,

- Number of lanes
- Traffic flow
- Speed of vehicle
- Lag suffered by vehicles during startup
- Similar road, intersection, and the same distance

We ran both simulations for an hour and recorded the number of vehicles that passed through the junction over three minutes three-minute periods in each system.

TABLE I. Total Cars Passed in the Traditional System

| No | Time | Lane 1 | Lane 2 | Lane 3 | Lane 4 | Total |
|---|---|---|---|---|---|---|
| 1 | 3 | 67 | 36 | 48 | 75 | 226 |
| 2 | 6 | 56 | 44 | 60 | 59 | 219 |
| 3 | 9 | 76 | 57 | 34 | 56 | 223 |
| 4 | 12 | 47 | 68 | 63 | 52 | 230 |
| 5 | 15 | 87 | 31 | 51 | 43 | 212 |
| 6 | 18 | 42 | 77 | 48 | 55 | 222 |
| 7 | 21 | 65 | 69 | 46 | 44 | 224 |
| 8 | 24 | 42 | 71 | 55 | 59 | 227 |
| 9 | 27 | 33 | 82 | 66 | 33 | 214 |
| 10 | 30 | 52 | 63 | 61 | 41 | 217 |
| 11 | 33 | 47 | 67 | 72 | 40 | 226 |
| 12 | 36 | 79 | 45 | 53 | 49 | 226 |
| 13 | 39 | 83 | 49 | 51 | 42 | 225 |
| 14 | 42 | 62 | 72 | 66 | 11 | 211 |
| 15 | 45 | 93 | 37 | 51 | 33 | 214 |
| 16 | 48 | 97 | 42 | 25 | 65 | 229 |
| 17 | 51 | 106 | 32 | 43 | 37 | 218 |
| 18 | 54 | 56 | 77 | 46 | 38 | 217 |
| 19 | 57 | 65 | 79 | 53 | 28 | 225 |
| 20 | 60 | 70 | 75 | 49 | 30 | 224 |

TABLE II. Total Cars Passed in the Proposed System

| No | Time | Lane 1 | Lane 2 | Lane 3 | Lane 4 | Total |
|---|---|---|---|---|---|---|
| 1 | 3 | 87 | 56 | 72 | 63 | 278 |
| 2 | 6 | 73 | 68 | 66 | 75 | 282 |
| 3 | 9 | 95 | 55 | 68 | 69 | 287 |
| 4 | 12 | 61 | 78 | 72 | 64 | 275 |
| 5 | 15 | 74 | 76 | 60 | 69 | 279 |
| 6 | 18 | 88 | 63 | 81 | 52 | 284 |
| 7 | 21 | 97 | 67 | 66 | 58 | 288 |
| 8 | 24 | 81 | 77 | 52 | 71 | 281 |
| 9 | 27 | 102 | 64 | 62 | 51 | 279 |
| 10 | 30 | 107 | 59 | 51 | 58 | 275 |
| 11 | 33 | 85 | 82 | 58 | 52 | 277 |
| 12 | 36 | 97 | 58 | 66 | 61 | 282 |
| 13 | 39 | 77 | 71 | 81 | 56 | 285 |
| 14 | 42 | 73 | 75 | 71 | 68 | 287 |
| 15 | 45 | 91 | 70 | 59 | 67 | 287 |
| 16 | 48 | 80 | 84 | 71 | 46 | 281 |
| 17 | 51 | 66 | 92 | 72 | 46 | 276 |
| 18 | 54 | 78 | 75 | 76 | 52 | 281 |
| 19 | 57 | 107 | 55 | 61 | 57 | 280 |
| 20 | 60 | 82 | 69 | 60 | 69 | 280 |

TABLE III. Comparison of Passing Cars in 2 Models

| No | Time | Total Car Passed(Current) | Total Car Passed(Proposed) | Difference |
|---|---|---|---|---|
| 1 | 3 | 226 | 278 | 52 |
| 2 | 6 | 219 | 282 | 63 |
| 3 | 9 | 223 | 287 | 64 |
| 4 | 12 | 230 | 275 | 45 |
| 5 | 15 | 212 | 279 | 67 |
| 6 | 18 | 222 | 284 | 62 |
| 7 | 21 | 224 | 288 | 64 |
| 8 | 24 | 227 | 281 | 54 |
| 9 | 27 | 214 | 279 | 65 |
| 10 | 30 | 217 | 275 | 58 |
| 11 | 33 | 226 | 277 | 51 |
| 12 | 36 | 226 | 282 | 56 |
| 13 | 39 | 225 | 285 | 60 |
| 14 | 42 | 211 | 287 | 76 |
| 15 | 45 | 214 | 287 | 73 |
| 16 | 48 | 229 | 281 | 52 |
| 17 | 51 | 218 | 276 | 58 |
| 18 | 54 | 217 | 281 | 64 |
| 19 | 57 | 225 | 280 | 55 |
| 20 | 60 | 224 | 280 | 56 |
| Total | | 4429 | 5624 | 1195 |

From Table I, we get the number of cars passing through the intersection every 3 minutes for 1 hour in the traditional system.

The result is shown in Fig. 7, where the number of vehicles passing through the intersection is between 210 and 230.

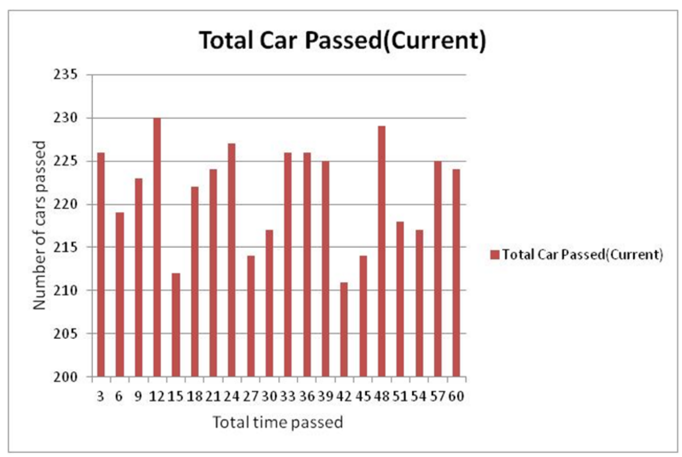


Fig. 7. Number of cars passed in the traditional system

From Table II, we get the number of cars passing through the intersection every 3 minutes for 1 hour in our proposed system.

The result is shown in Fig. 8, where between 270 and 290 automobiles traveled past the crossing.

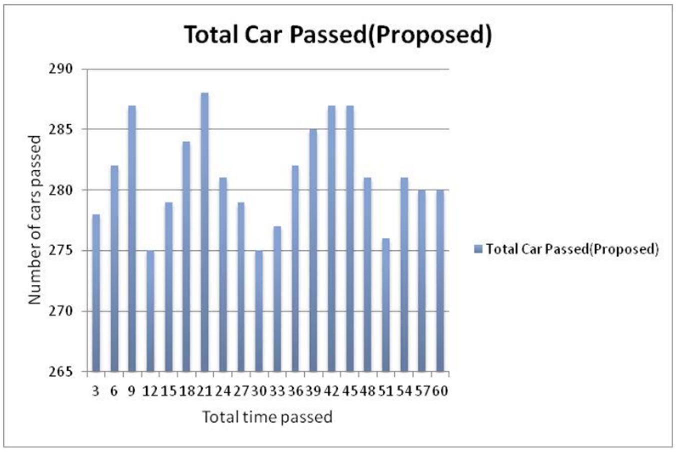


Fig. 8. Number of cars passed in the proposed system

From Table III, we can see the comparison of the result of the simulation we have run for both systems. The result, in this case, is quite satisfactory.

The result is shown in Fig. 9, where 40-80 more vehicles are being passed through the intersection in our proposed system compared to the traditional traffic system.


Fig. 9. Comparison of the number of cars passed in both systems

Here are 20 different results of 1 hour of simulation of both systems. Every 3 minutes, the traditional system lets 210-230 vehicles pass through the intersection whereas our proposed system is letting 270-290 vehicles pass through the intersection, which is a noticeable improvement.

TABLE IV. Comparison between 2 Models

| Parameters | Traditional system | Proposed system |
|---|---|---|
| Average cars passed per minute | 74 | 94 |
| Total cars passed in 1 hour | 4429 | 5624 |

By comparing the simulations while maintaining the same set of conditions, the outcome of Table IV was obtained.

As we can see, keeping all the conditions alike, the current system was able to pass 4429 vehicles total in 1 hour, whereas our proposed system was able to pass 5624 vehicles in same amount of time, which is 1195 more vehicles than current system.

Thus, our proposed system will increase the performance by 26.98126%

## V. Conclusion

In conclusion, the suggested program guarantees that, by adaptively modifying the green signal timing according to the density at the light, the route with more vehicles is given a green light time for a longer period in comparison to the route with less traffic. Due to fewer unnecessary delays, less traffic, and shorter wait times, there will be less fuel use and pollution.

In regards to the number of vehicles passing the intersection, the system performs around 23 percent better than the current method, which is a substantial improvement. With more calibration and authentic surveillance data for the training set, this system has the potential to perform even more effectively. Additionally, the suggested method has some benefits over other intelligent traffic management systems that are already used, like Pressure Pads and Infrared Sensors. The system may be installed for a very low cost because it uses CCTV footage from traffic signal cameras and typically doesn't need any new hardware because these cameras are already present at intersections with considerable traffic.

There might only be a small amount of alignment required. Maintenance expenses are also decreased when compared to other traffic enforcement methods like "pressure mats", which frequently face stress for their installation on roadways where these are continuously imposed to tremendous stress. To allow improved traffic control, the proposed method can therefore be connected with Surveillance cameras in large cities

## VI. Future Work

To improve traffic management and reduce congestion, the project could be extended further to include features like these:

*1) Detecting violations of traffic rules:* Vehicles that are breaking traffic rules can be identified easily by video streaming and capturing images of vehicles with number plates. It is feasible to recognize the automobiles that are skipping red lights in the picture or video stream by establishing a transgression barrier and recording the plate number of the image and whether that line has been crossed when the light is red. Similarly, lane changing of vehicles might also be detected. These can be accomplished through image processing methods or backdrop removal.

*2) Detecting violence or accidents on road:* Accidents can be detected easily and necessary steps can be taken quickly. Intersections commonly see catastrophic collisions as a result of the presence of numerous hazardous collapses, like crashes due to angular and turning left. To safeguard property and lives while simultaneously minimizing traffic and delays, it is crucial to accurately and quickly identify accidents at crossings. The list of automobiles that remain immobile for a lengthy amount of time in an inhospitable area, such as the center of the road, might be modified to exclude parked cars from this restriction.

Also if any occurrence happens by any vehicle, it can be detected and tracked. Also, the probability of the vehicle taking a specific route can be predicted in advance.

*3) Detecting emergency vehicles:* Vehicles like ambulances or fire service can be detected and the timer can be adapted accordingly to make a quick passage for the emergency vehicle. The model can be programmed to recognize emergency vehicles in addition to other types of cars, allowing it to adjust timers to give them precedence and enable them to cross the signal as soon as possible.

## REFERENCES


[1] Khushi, "Smart Control of Traffic Light System using Image Processing", 2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC), 2017, PP. 99–103.

[2] Soman, Renjith and Radhakrishnan, K, "Traffic Light Control and Violation Detection Using Image Processing", Traffic, vol. 8, No. 4, 2018.

[3] Chabchoub, Abdelkader and Hamouda, Ali and Al-Ahmadi, Saleh and Cherif, Adnen, "Intelligent traffic light controller using fuzzy logic and image processing", International Journal of Advanced Computer Science and Applications (IJACSA), vol. 12, No. 4, 2021.

[4] Suhweil, Yazan and Al Yaman, Musa and others, "Smart controlling for traffic light time",2017 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), IEEE, 2017, PP. 1–5.

[5] Al-Majhad, Hailah Ghanem and Bramantoro, Arif and Syamsuddin, Irfan and Yunianta, Arda and Basori, Ahmad Hoirul and Prabuwono, Anton Satria and Barukab, Omar M, "Intelligent traffic light controller using fuzzy logic and image processing", International Journal of Advanced Computer Science and Applications (IJACSA), vol. 9, No. 4, 2018.

[6] Agrawal, Aditi and Paulus, Rajeev, "Smart Intersection Design for Traffic, Pedestrian and Emergency Transit Clearance using Fuzzy Inference System", International Journal of Advanced Computer Science and Applications (IJACSA), vol. 12, No. 3, 2021.

[7] Adbeb, Tesfanesh and Wu, Di and Ibrar, Muhammad, "Software-defined networking (SDN) based VANET architecture: Mitigation of traffic congestion", International Journal of Advanced Computer Science and Applications (IJACSA), vol. 11, No. 3, 2020.

[8] Shinde, Ms Saili and Jagtap, SR, "Intelligent Traffic Management Systems", A Review, vol. 2, No. 9, 2016.

[9] Navarathna, Pramathi J and Malagi, Vindhya P, "Artificial intelligence in smart city analysis",2018 International conference on smart systems and inventive technology (ICSSIT), IEEE, 2018, PP. 44–47.

[10] de Frías, Carlos Justo and Al-Kaff, Abdulla and Moreno, Francisco Miguel and Madridano, Ángel and Armingol, José María, "Artificial intelligence in smart city analysis",2020 IEEE Intelligent Vehicles Symposium (IV), IEEE,vol. 10, 2020, PP. 33–38.

[11] Manjón Prado, Javier, "Using Unreal Engine as an engineering tool for traffic simulation and analysis",Universitat Politècnica de Catalunya, vol. 10, 2020.

[12] Karimi, Abolfazl and Duggirala, Parasara Sridhar, "Formalizing traffic rules for uncontrolled intersections",2020 ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPS), IEEE, 2020, PP. 41–50.

[13] Golubev, Kirill and Zagarskikh, Aleksandr and Karsakov, Andrey, "A framework for a multi-agent traffic simulation using combined behavioural models", Procedia Computer Science, Elsevier, vol. 136, 2018, PP. 443–452.