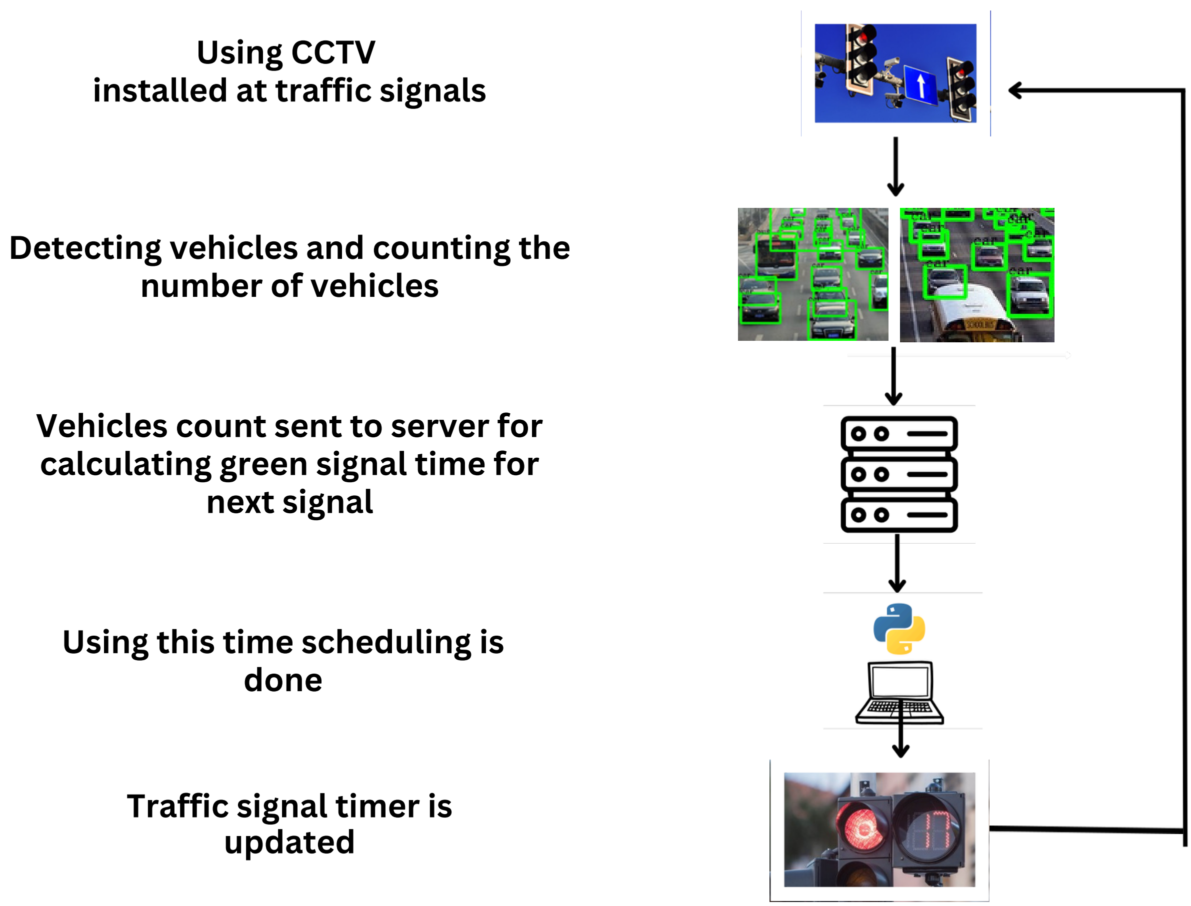
Smart System for Traffic Flow Optimization And Congestion Management

Implementation Details

Proposed System Overview

Our proposed system uses the CCTV cameras installed at traffic junctions to count the number of vehicles using vehicle detection. This system can be broken down into 3 modules : Vehicle Detection module, Signal Switching Algorithm, and Simulation module. The Signal Switching Algorithm determines green signal times for each arm by averaging the traffic density observed over previous cycles. The red signal times are updated accordingly. The green signal time is restricted to a maximum and minimum value in order to avoid starvation of a particular lane. A simulation is also developed to demonstrate the system's effectiveness and compare it with the existing static system.



Vehicle Detection Module

Our proposed system uses YOLO(You only look once) for vehicle detection, followed by vehicle tracking, where each detected vehicle is assigned a unique ID to prevent double counting. After the detection phase, our tracker monitors the movement of vehicles, ensuring accurate counting and analysis of traffic flow. 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.

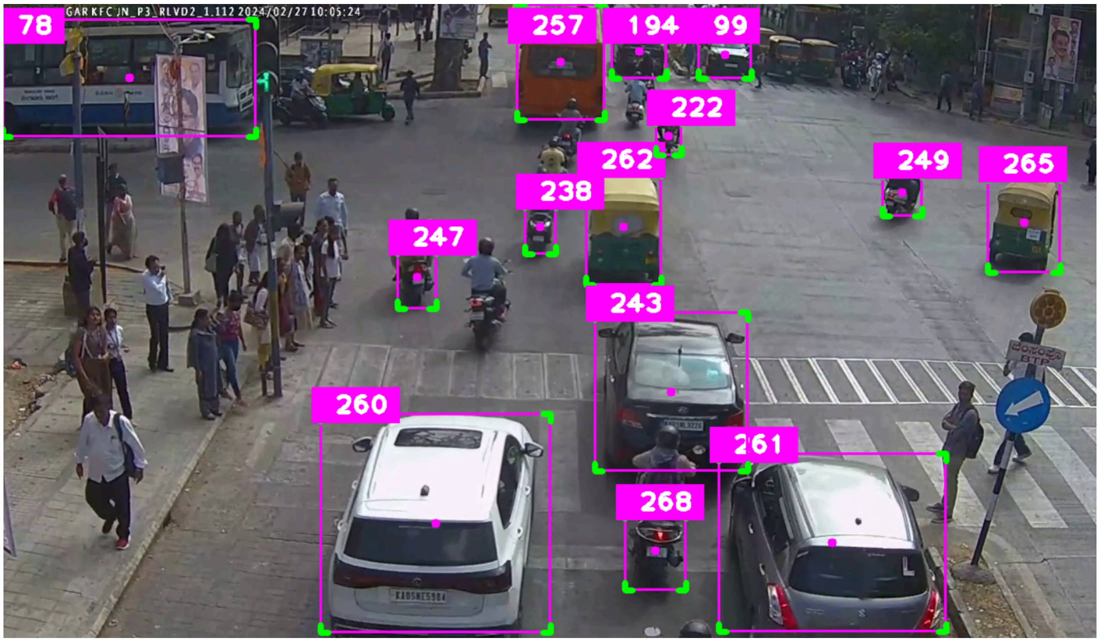
-The dataset for training the model was prepared by scraping images from google and labelling them manually using LabelIMG, a graphical image annotation tool.

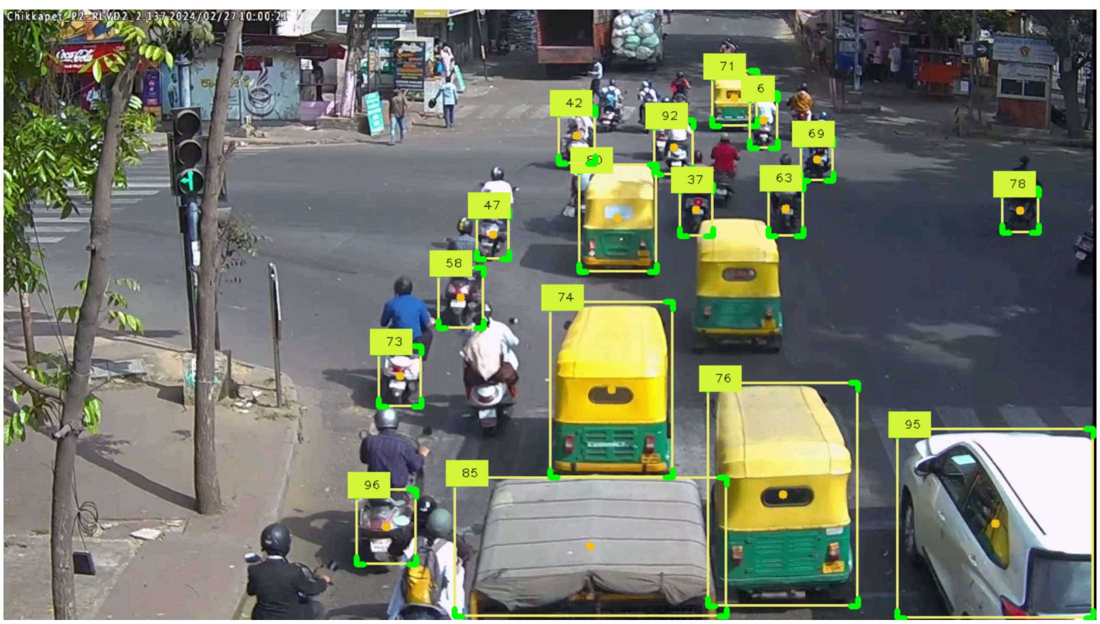
-Then the model was trained using the pre-trained weights downloaded from the YOLO website. The configuration of the .cfg file used for training was changed in accordance with the specifications of our model. The number of output neurons in the last layer was set equal to the number of classes the model 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:





Signal 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

Working of the algorithm
When the algorithm is first run, the default time is set for the first signal of the first cycle and the 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 5 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. 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 5 seconds. This gives the system a total of 10 seconds 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

$$\textbf{Green Signal Time} = \textbf{Density factor} \times \textbf{Predicted taffic}_{\text{arm wise}}$$

$$\textbf{Predicted traffic} = \frac{\sum_{n=-1}^{-3} \textbf{Vehicle Count}}{3}$$

$$\textbf{Density factor} = \frac{\text{max signal time}}{\textbf{max Traffic on the lane}}$$

where:
GST is green signal time

- • **No Of Vehicles Of Class** 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
- • **Number Of Lanes** 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

Mitigating Traffic Bottlenecks Caused by Large Vehicle Obstructions: Smart Solutions for Efficient Flow

Additional Innovation : Bottleneck Situation

When a large vehicle, such as a truck or bus, is parked inappropriately—either on a narrow street, close to an intersection, or in a designated no-parking zone—it can create a significant bottleneck in traffic. *Here's how and why it happens:*

Why the Bottleneck Occurs:

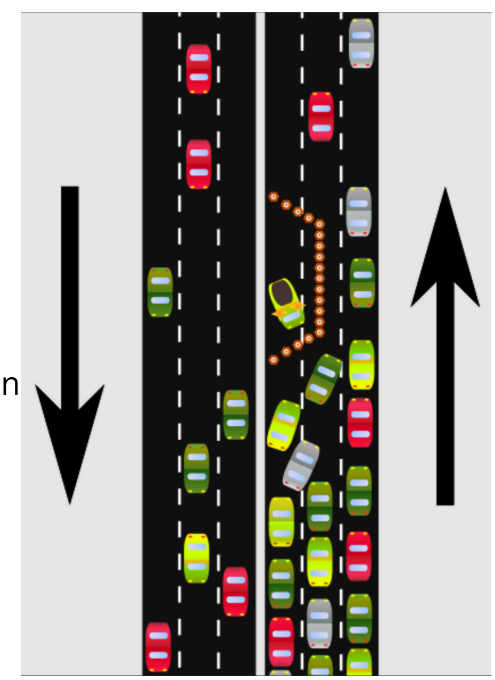
1. Reduced Lane Width: Large vehicles occupy more space than regular cars. When parked on the side of the road or in a place not meant for parking, they reduce the available lane width. This forces vehicles to either slow down significantly or merge into adjacent lanes, causing traffic to back up.

2. Disruption in Flow: Traffic patterns are designed with specific lane capacities in mind. When a lane is partially blocked, it interrupts the smooth flow of vehicles, causing congestion. Drivers may need to take turns merging into one lane, which slows down all traffic in the area.

3. Icreased Safety Risks: Large vehicles obstructing traffic can force smaller vehicles to make sudden lane changes or attempt unsafe maneuvers, further slowing traffic and increasing the risk of accidents.

How It Affects Traffic:

Queue Formation: Vehicles line up behind the obstruction, reducing throughput in the area. This causes a queue, which often stretches into adjacent streets, spreading the congestion.



Wider Traffic Impact: Traffic congestion caused by a large parked vehicle can have a ripple effect, slowing down traffic not only in the immediate area but also in surrounding intersections or roads as more vehicles are funneled into fewer lanes.

Increased Wait Time: If the obstruction is near a traffic signal, the reduced capacity of the road can mean that fewer vehicles pass through each green light, further extending wait times and causing frustration for drivers.

How Smart Traffic Management Systems Can Help:

- *Real-Time Detection and Alerts:*

Why: Detecting large vehicles in real-time can prevent the situation from escalating. Authorities can be immediately informed of illegally parked or obstructive vehicles.
How: Using *Avinya's solution* CCTV cameras smart traffic systems can detect large vehicles parked in problematic locations in red on a digital display. Once detected, an alert can be sent to traffic enforcement or to the driver directly. This system reduces manual monitoring and ensures faster response times.

- *Adaptive Traffic Signals:*

Why: Traffic signals can be dynamically adjusted based on real-time traffic conditions, allowing traffic to flow more smoothly around obstructions.
How: If a large vehicle is causing a bottleneck, traffic signals can be adapted to give more green light time to the lanes affected by the obstruction, clearing congestion faster. For example, if a lane is blocked, the signals for adjacent lanes could be extended to handle more vehicles while allowing less time for the obstructed lanes.

Benefits of Addressing Large Vehicle Parking Bottlenecks:

- *Increased Traffic Efficiency:* By removing obstructions quickly, traffic can resume its normal flow, reducing delays for all road users

- *Safety Improvements:* Reducing the risks of accidents caused by obstructed sight lines or sudden lane changes improves overall road safety.

- *Time Savings:* Faster response times and adaptive systems lead to less time spent in traffic for drivers, reducing frustration and improving commuter satisfaction.

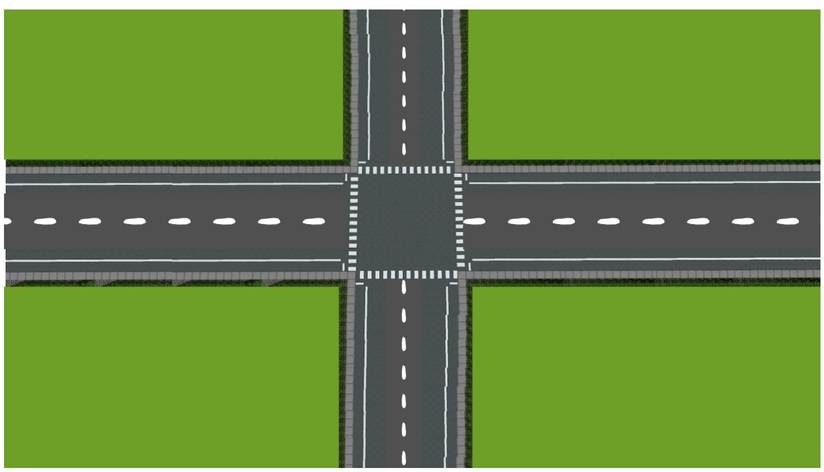
Bottlenecks can be recurring or nonrecurring. According to a government report, **40%** of road delays are due to recurring bottlenecks.

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. Additionally, we have added pedestrian crossing for 10 seconds after the completion of every cycle.

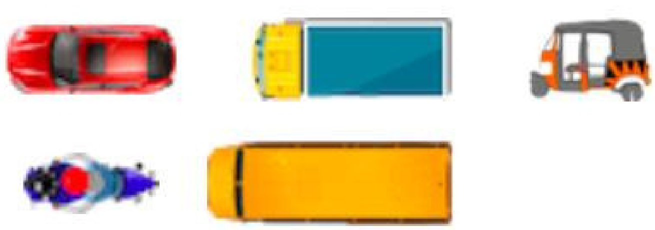
Key steps in development of simulation

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

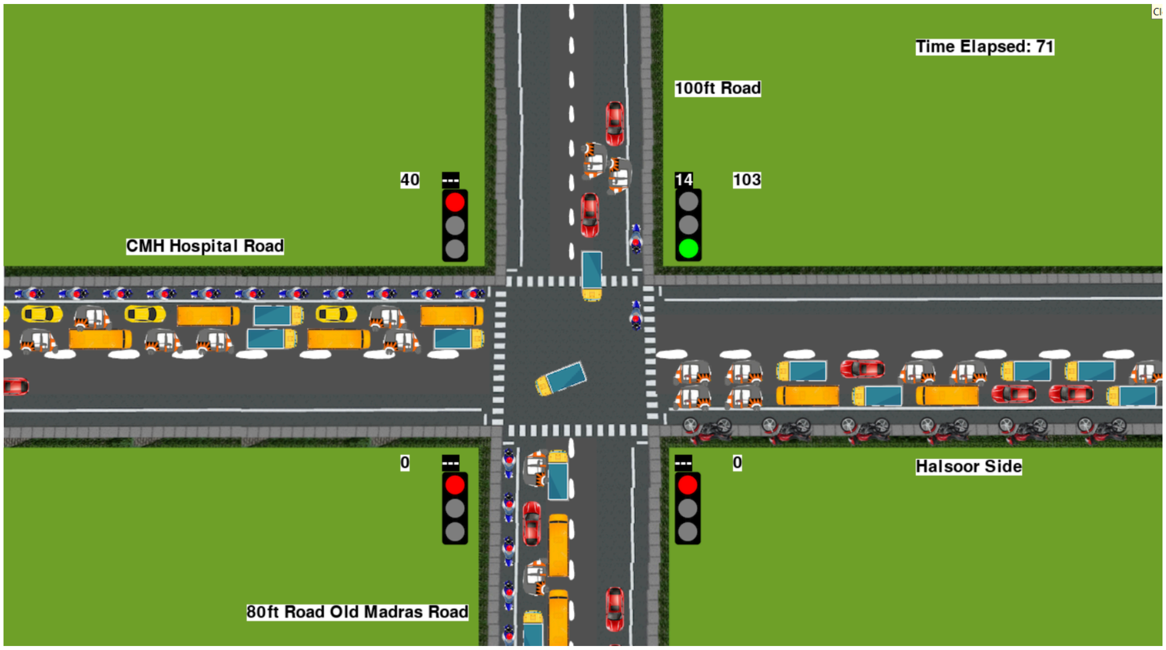


2. Gathered top-view images of car, bike, bus, truck, and rickshaw. Resized them.

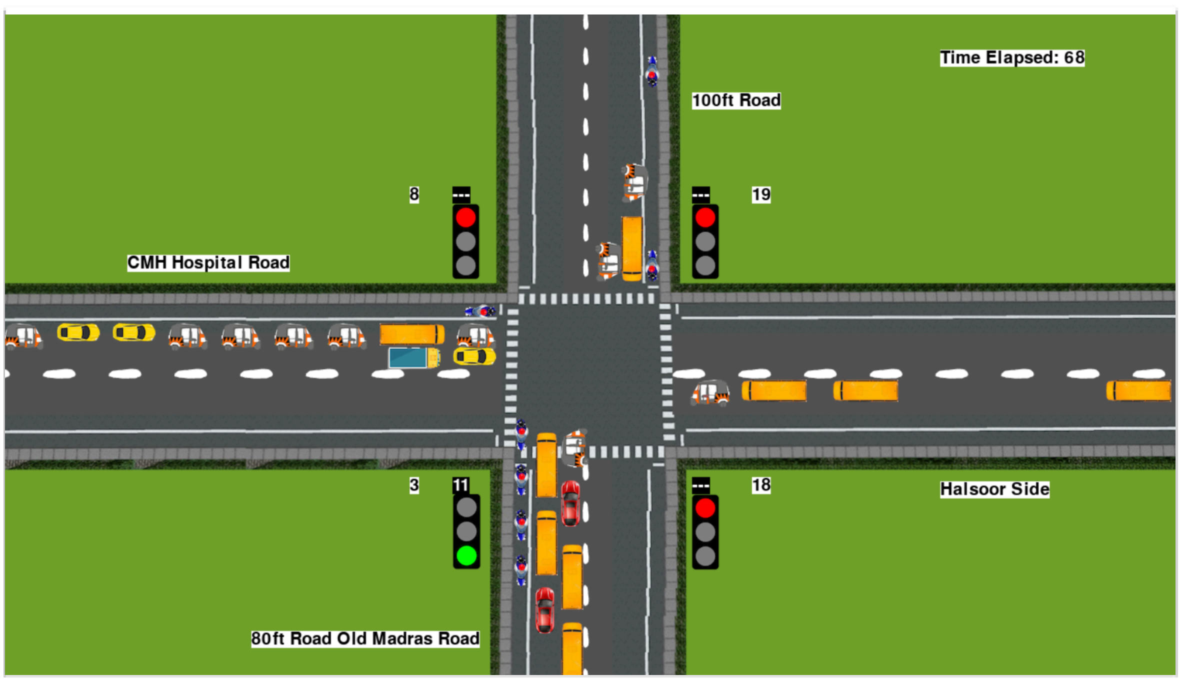
3.



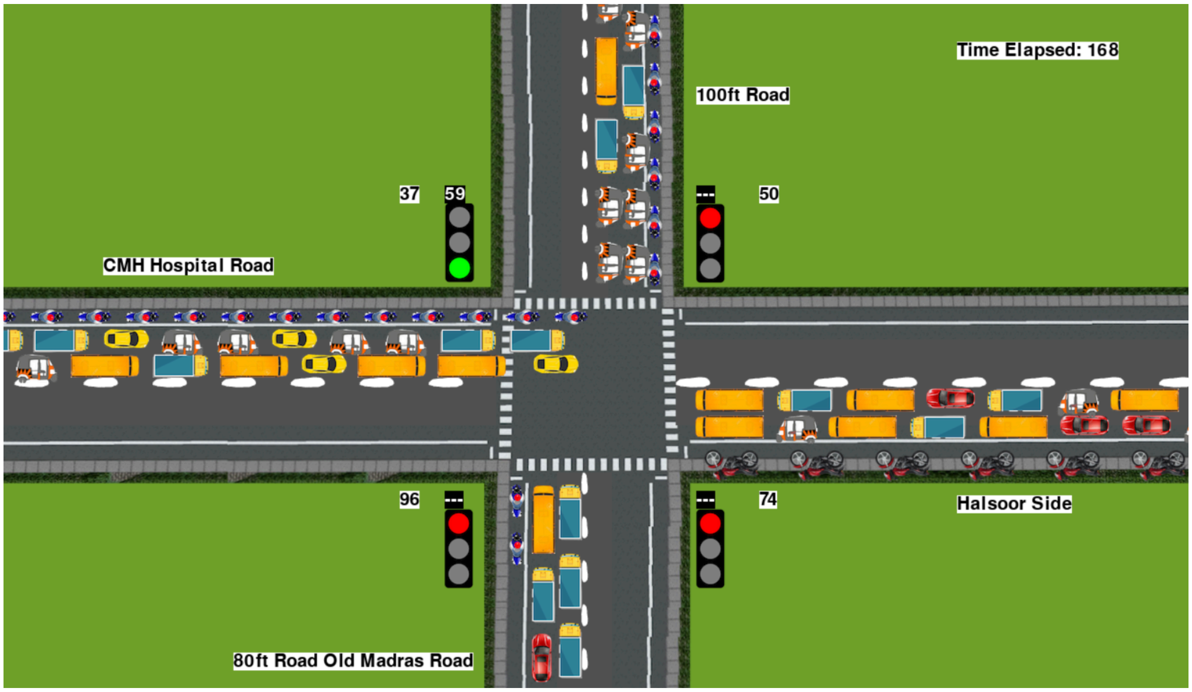
Following are some images of the demo simulation:



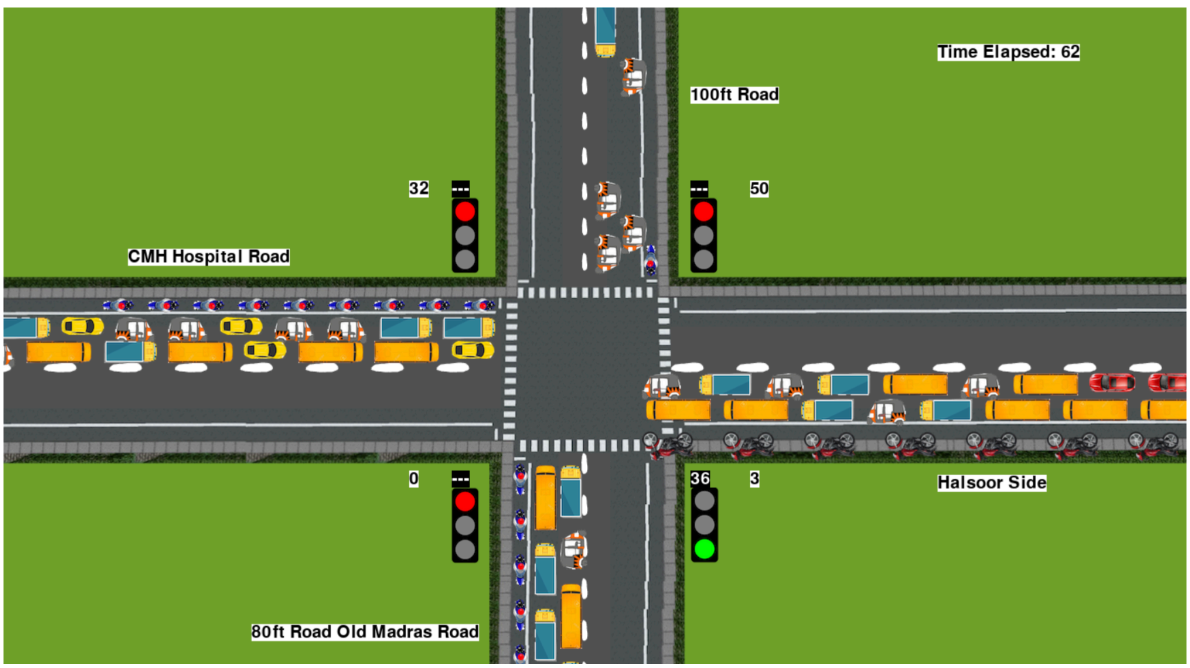
(iii): Simulation showing vehicles turning



(iv): Simulation showing green time of signal for vehicles moving up set to 11 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 11 seconds in this case.



(v): Simulation showing green time of signal for vehicles moving right set to 59 seconds according to the vehicles in that direction.



(vi): Simulation showing green time of signal for vehicles moving left set to 36 seconds according to the vehicles in that direction.

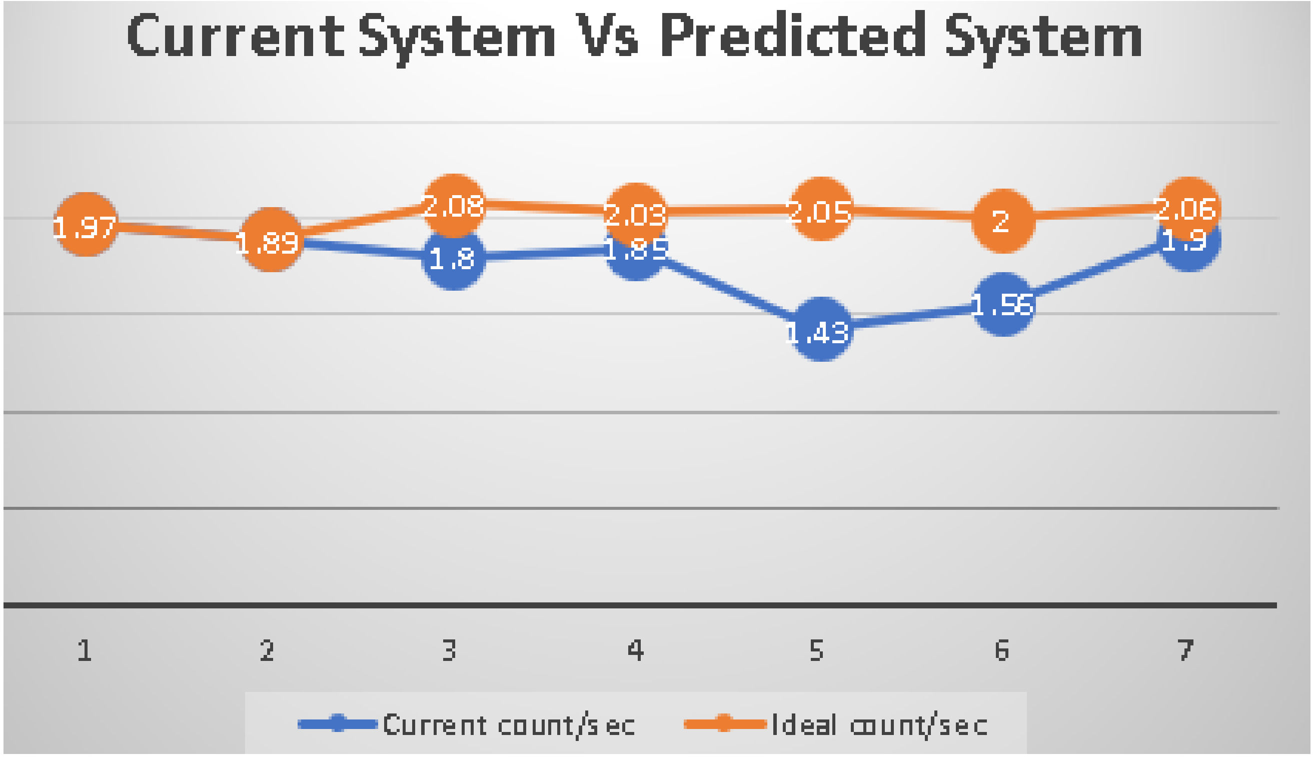
Results

We compare the vehicle count per unit time for each lane at the given road junction and present a comparison between the current system and the ideal situation with the same distribution.

The results obtained are stored in a tabulated format, where each cell represents the count per time of each lane for that particular cycle. Additionally, the total time and vehicle count for each cycle are recorded.

Current Situation (Indira Nagar)						
Cycle No.	Lane 1 (CMH ROAD) count/Time	Lane 2 (100ft road) count/Time	Lane 3 (OLD MADRAS	Lane 4 (HALSOOR) count/Time	Total count	Total Tim
1	65/35	84/50	70/30	87/40	306	155
2	53/35	88/50	85/30	68/40	294	155
3	62/35	92/50	62/30	72/40	288	155
4	69/35	83/50	60/30	68/40	280	155
5	51/35	73/50	40/30	59/40	223	155
6	76/35	82/50	30/30	50/40	238	155
7	67/35	98/50	63/30	65/40	293	155
					1922	1085

Predicted System (Indra Nagar)						
Cycle No.	Lane 1 (CMH ROAD) count/Time	Lane 2 (100ft road) count/Time	Lane 3 (OLD MADRAS ROAD)	Lane 4 (HALSOOR) count/Time	Total Count	Total Tim
1	65/35	84/50	70/30	87/40	306	155
2	53/35	88/50	85/30	68/40	294	155
3	59/36	83/46	77/27	77/35	296	142
4	61/38	88/46	72/25	73/33	295	145
5	60/37	87/48	69/25	69/30	285	139
6	60/37	82/45	54/20	66/29	263	131
7	65/35	79/44	44/15	62/27	250	121
					1989	833



As seen in the above tables, we observe that the same number of vehicles are processed in less time in the proposed system compared to the current situation. Previously, it took approximately 1085 seconds over seven cycles to process a similar volume of vehicles. However, in the ideal situation, we achieve the same throughput within 833 seconds for the same seven cycles. This implies that our proposed system is 20% more efficient compared to the the current system.

Conclusion

Thus, 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 longer duration of time as compared to the direction with lesser traffic. This will lower the unwanted delays, and reduce congestion and waiting time which in turn will reduce the fuel consumption and pollution.

According to simulation results, the system shows about 20% 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.