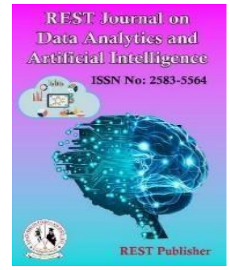
REST Journal on Data Analytics and Artificial Intelligence
Vol: 4(1), March 2025
REST Publisher; ISSN: 2583-5564
Website: http://restpublisher.com/journals/jdaai/
DOI: https://doi.org/10.46632/jdaai/4/1/52



Leveraging Deep Learning for Traffic Management using Existing System

***Tanish Vardhineni, Yashwanth Reddy, T Arun Kumar, Ravi Kumar**
Anurag University, Hyderabad, Telangana - 500088, India.
**Corresponding Author Email: 21eg107a53@anurag.edu.in*

Abstract. *Urban traffic congestion is a pressing issue that worsens with the rise of vehicular density in cities. Traditional traffic management systems, based on fixed signal timings, lack the adaptability needed to handle the dynamic and often unpredictable nature of traffic flow. This project introduces an AI-based traffic signal control system that leverages real- time video data from CCTV cameras to optimize traffic signal timings based on actual traffic conditions dynamically. By integrating computer vision techniques and machine learning models, the system can calculate traffic density in real time and adjust traffic signal timings to reduce waiting times, fuel consumption, and pollution. The project aims to improve urban traffic management by providing a smart, adaptive, and scalable solution.*

Keywords: *Industrial-Internet-of-things-(IIOT), Deep Learning, Convolution Neural Network, Traffic Signals, Traffic Lights, cameras.*

1. INTRODUCTION

With the increasing number of vehicles in urban areas, many road networks are facing problems with the capacity drop of roads and the corresponding Level of Service. Many traffic-related issues occur because of traffic control systems at intersections that use fixed signal timers. They repeat the same phase sequence and its duration with no changes. Increased demand for road capacity also increases the need for new solutions for traffic control that can be found in the field of Intelligent Transport Systems. Let us take the case study of Mumbai and Bangalore. Traffic flow in Bangalore is the worst in the world while Mumbai is close behind in fourth position, according to a report detailing the traffic situation in 416 cities across 57 countries. In Bangalore, a journey during rush hour takes 71% longer. In Mumbai, it is 65% longer [1-5]. Three standard methods for traffic control are being used currently: Manual Controlling: As the name suggests, it requires manpower to control the traffic. The traffic police are allotted for a required area to control traffic. The traffic police carry signboards, sign lights, and whistle to control the traffic [6]. Conventional traffic lights with static timers: These are controlled by fixed timers. A constant numerical value is loaded in the timer. The lights are automatically switching to red and green based on the timer value [7]. Electronic Sensors: Another advanced method is placing some loop detectors or proximity sensors on the road. This sensor gives data about the traffic on the road. According to the sensor data, the traffic signals are controlled. These conventional methods face certain drawbacks [8]. The manual controlling system requires a large amount of manpower. As there is poor strength of traffic police, we cannot have them controlling traffic manually in all areas of a city or town. So a better system to control the traffic is needed. Static traffic controlling uses a traffic light with a timer for every phase, which is fixed and does not adapt according to the real-time traffic on that road [9]. While using electronic sensors i.e., proximity sensors or loop detectors, the accuracy and coverage are often in conflict because the collection of high-quality information is usually based on sophisticated and expensive technologies, and thus limited budget will reduce the number of facilities. Moreover, due to the limited effective range of most sensors, the total coverage on a network of facilities usually requires a lot of sensors. In recent years, video monitoring and surveillance systems have been extensively used in traffic management for security, ramp metering, and providing information and updates to travelers in real-time. The traffic density estimation and vehicle classification can also be achieved using video monitoring systems, which can then be used to control the timers of the traffic signals so as to optimize traffic flow and minimize congestion. Our proposed system aims to design a traffic light controller based on Computer Vision that can adapt to the current traffic situation. It uses live images from the CCTV cameras at traffic junctions for real-time traffic density calculation by detecting the number of vehicles at the signal and setting the green signal time accordingly. The vehicles are classified as a car, bike, bus/truck, or rickshaw to obtain an accurate estimate of the green signal time. It uses YOLO in order to detect the number of vehicles and then set the timer of the traffic signal according to vehicle density in the corresponding direction. This helps to optimize the green signal times, and traffic is cleared at a much faster rate than a static system, thus reducing the unwanted delays, congestion, and waiting time, which in turn will reduce the fuel consumption and pollution [10-13].

2. BACKGROUND

Existing System: Conventional traffic management systems either rely on manual oversight or use fixed- timing algorithms [14-16]. These systems are typically ineffective in managing modern, high-volume traffic flows due to their lack of flexibility. Key drawbacks of traditional systems include: Inflexibility: Fixed-timing traffic signals do not account for variable traffic volumes, resulting in inefficient flow control. Increased Idling: Vehicles waste fuel while waiting for green signals, even when there is little or no traffic in opposing directions. Lack of Real-Time Data: Traditional systems do not use real-time traffic data, leading to poor traffic management, especially in rapidly changing traffic situations such as emergencies or high-traffic periods.

Limitations Existing System: The rising number of vehicles, combined with limited road infrastructure, has made traffic congestion a significant problem in urban areas [17]. Traffic jams result in: Longer commuting times. Increased stress and frustration for drivers. Higher fuel consumption, leading to elevated operational costs. Increased emissions and worsening air quality.

Gaps Identified: Based on the limitations mentioned above, the following gaps have been identi5ied in the existing traf5ic management systems: Existing systems lack the capability to adapt to rapidly changing traffic conditions in real-time, which is crucial in high-density urban environments. The fixed nature of current systems is inefficient, particularly during peak hours, emergencies, or traffic surges [18]. Manual and sensor-based systems often lead to underutilization or overutilization of road infrastructure. For example, lanes with low traffic might get more green light time than necessary, while busy lanes suffer from extended red-light durations. Current systems do not leverage advanced machine learning or computer vision technologies for vehicle detection, classification, or density estimation, which would allow for a more intelligent and adaptive traffic control mechanism [19]. Hardware-based systems are costly to install and maintain. Additionally, reliance on manual traffic control or sensor-based technologies without automated learning systems can increase operational costs and reduce efficiency.

Problem Statement: The rapid growth in vehicular density in urban areas necessitates a more responsive and adaptable traf5ic management system. Current traf5ic control methods, based on 5ixed-time or manual systems, are incapable of ef5iciently managing the ever-changing traf5ic conditions, leading to prolonged waiting times, wasted fuel, increased air pollution, and stress for commuters. There is a need for an AI-based traf5ic control system that utilizes real-time video data from CCTV cameras, along with machine learning models, to dynamically adjust signal timings based on actual traf5ic density. This system will optimize traf5ic 5low by minimizing congestion and reducing emissions, fuel consumption, and delays, offering a scalable and cost-effective solution for urban traf5ic management [20].

3. DESIGN

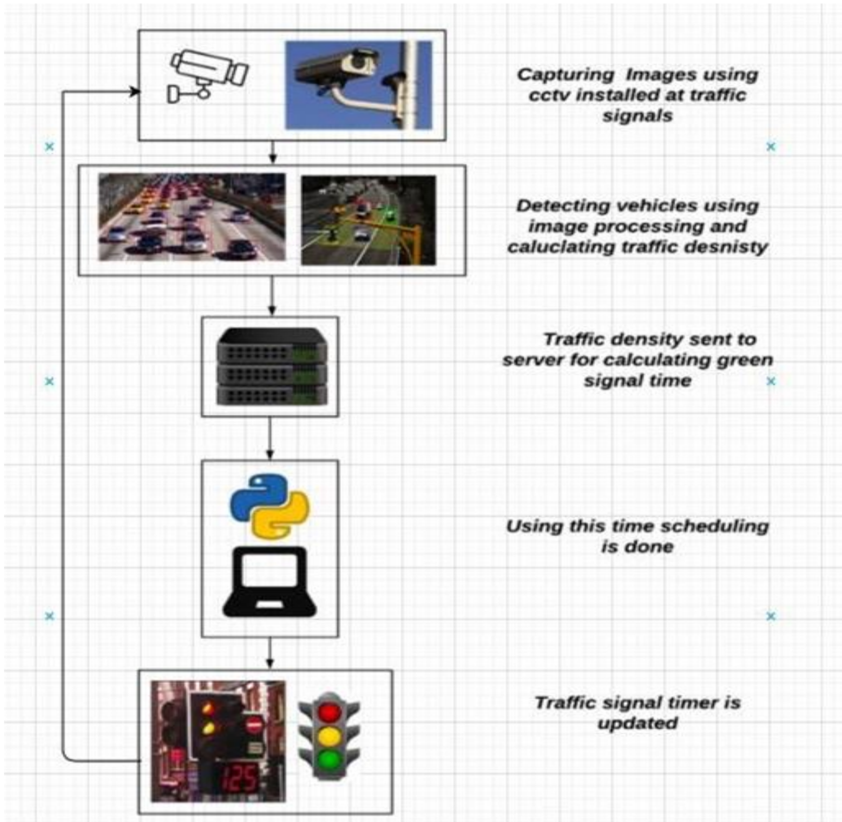


FIGURE 1. Model

4. IMPLEMENTATION & TESTING

A.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 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. Following are some images of the output of the Vehicle Detection Module:

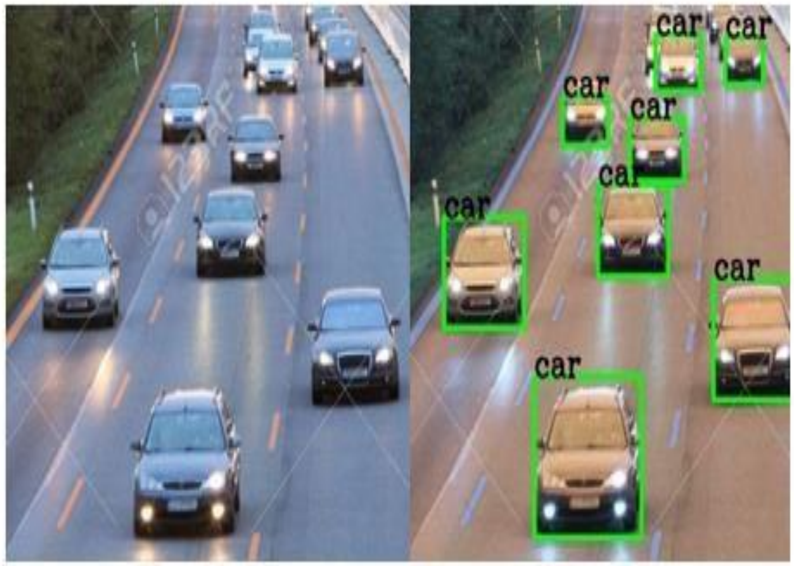


FIGURE 2. Vehicle Detection Model

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:

- 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
- Number of lanes
- Total count of vehicles of each class like cars, trucks, motorcycles, etc.
- Traffic density calculated using the above factors
- 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 [13]
- 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 [14]
- The minimum and maximum time limit for the green light duration - to prevent starvation

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:

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

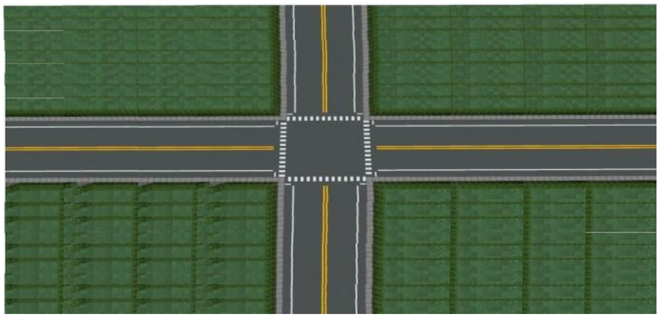


FIGURE 4. 4-way intersection

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

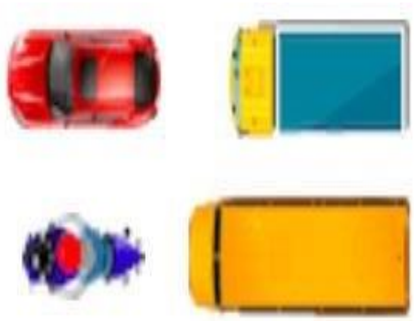


FIGURE 5. top-view images of Vehicles

- Rotated them for display along different directions.

FIGURE 6. Car 4-directions Image

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

FIGURE 7. Red, green, yellow Signals

- rendering the appropriate image of the signal depending on whether it is red, green, or yellow.

- 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.
- 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.
- 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 it does not crash into the bus.
- 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.
- displaying the number of vehicles that have crossed the signal.
- displaying the time elapsed since the start of the simulation.
- 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.
- 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.
- Vehicles turning and crossing the intersection in the simulation to make it more realistic.Following are some images of the final simulation:

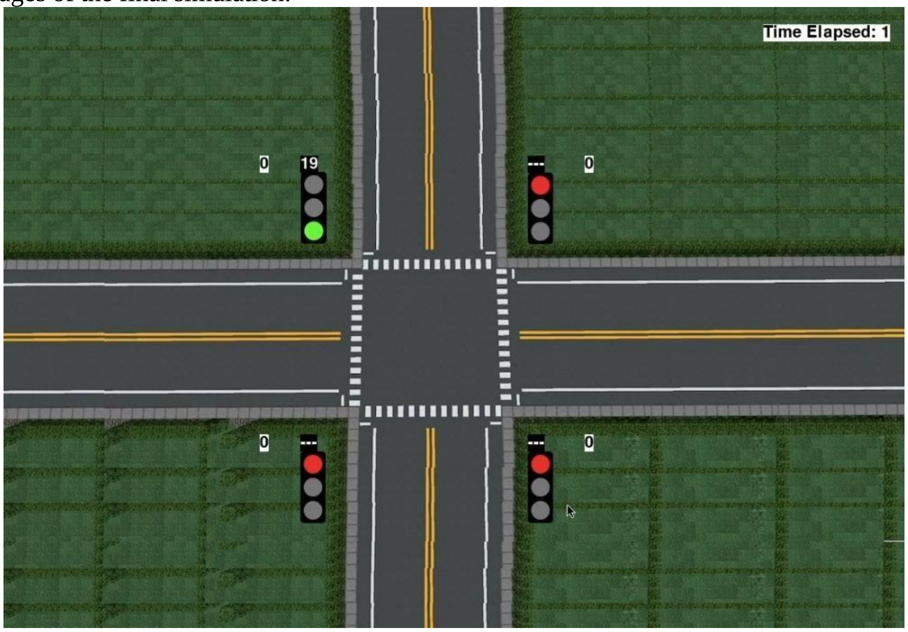


FIGURE 8. Simulation (i): just after start showing red and green lights, green signal time counting down from a default of 20 and red time of next signal blank. When the signal is red, we display a blank value till it reaches 10 seconds. The number of vehicles that have crossed can be seen beside the signal, which are all 0 initially. The time elapsed since the start of simulation can be seen on top right.

FIGURE 9. Simulation (ii) showing vehicles turning

FIGURE 9. Simulation (iii) 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.

.

FIGURE 10: Simulation (iv) 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.

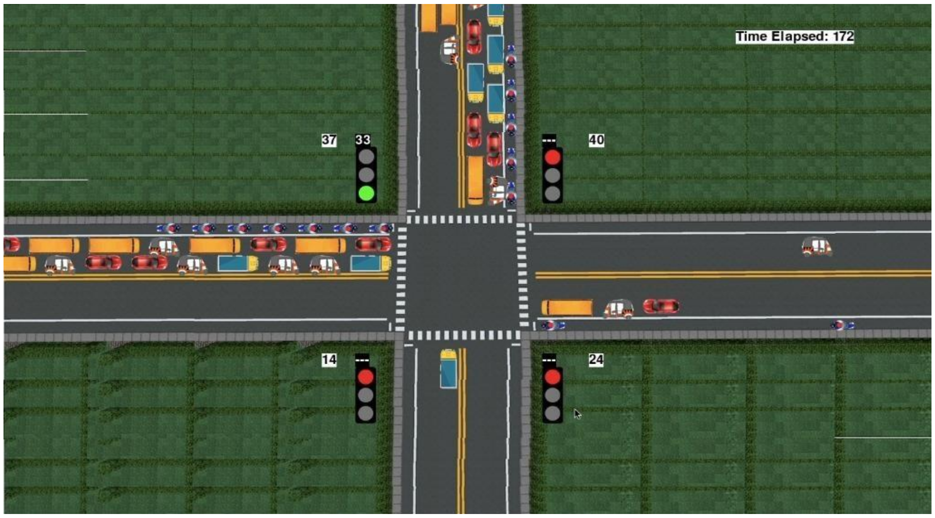


FIGURE 11: Simulation (v) showing green time of signal for vehicles moving right set to 33 seconds according to the vehicles in that direction.

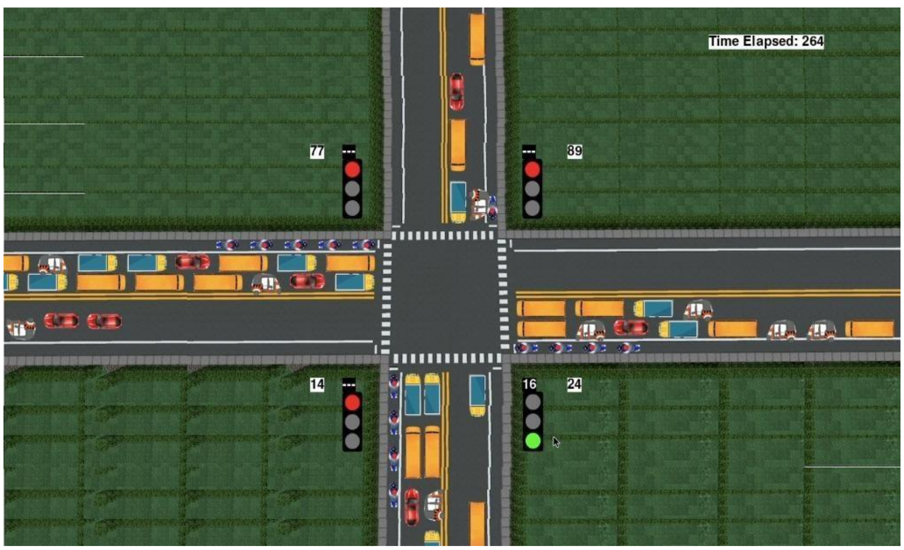


FIGURE 12: Simulation (vi) showing green time of signal for vehicles moving left set to 24 seconds according to the vehicles in that direction.

5. RESULT

Output: The traffic flow analysis system successfully detects vehicles in real time using live CCTV feeds and adjusts traffic signal timings based on calculated traffic density. The system outputs detailed vehicle counts, including categorization of vehicles such as cars, buses, and trucks, along with their movement trajectories within defined zones. It also generates real-time traffic density data, which is used to modify the green signal timings dynamically, resulting in optimized traffic flow. The system's performance was visually confirmed through a dashboard displaying live video feeds with vehicle detection overlays, traffic density graphs, and signal timing changes in real time. The reduction in traffic congestion at the monitored intersections is one of the key outputs, demonstrating the system's ability to adapt signal timings efficiently.

FIGURE 13. Output (i)

FIGURE 14. Output (ii)

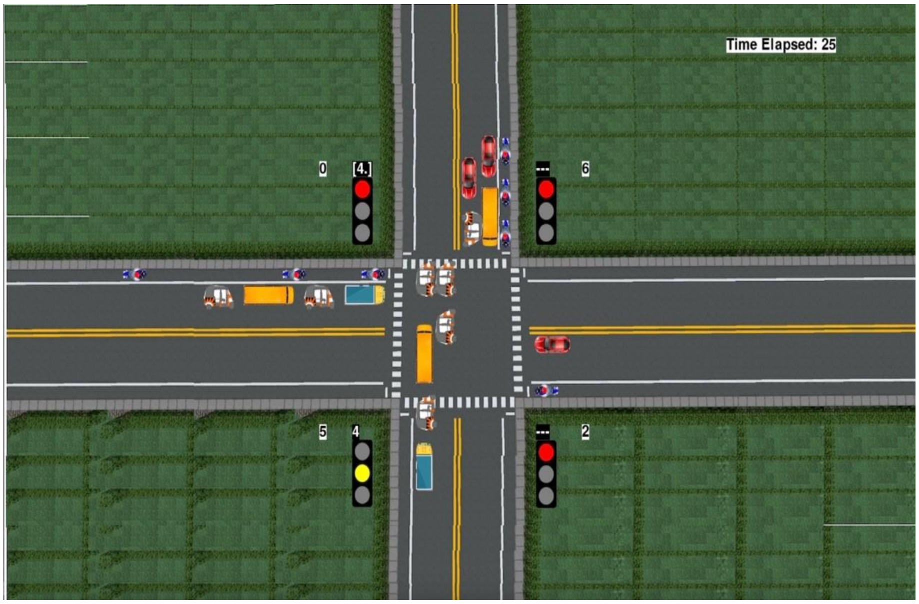


FIGURE 15. Output (iii)

6. CONCLUSION & FUTURE WORK

In 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 reduce congestion and waiting time, which in turn will reduce fuel consumption and pollution. According to simulation results, the system shows about 23% improvement over the current system in terms of the number of vehicles crossing the intersection, which is a significant improvement. With further calibration using real- life CCTV data for training the model, this system can be improved to perform even better. Moreover, the proposed system possesses certain advantages over the existing intelligent traffic control systems prevalent such as Pressure Mats and Infrared Sensors. The cost required to deploy the system is negligible as footage from CCTV cameras from traffic signals is used, which requires no additional hardware in most cases, as intersections with heavy traffic are already equipped with such cameras. Only minor alignment may need to be performed. The maintenance cost also goes down as compared to other traffic monitoring systems such as pressure mats that normally suffer wear and tear due to their placement on roads where they are subjected to immense pressure constantly. Thus, the proposed system can thus be integrated with the CCTV cameras in major cities in order to facilitate better management of traffic. The project can be further expanded to include the following functionalities to enhance traffic management and bring down congestion:

- Identification of vehicles violating traffic rules: The vehicles running red lights can be identified in an image or a video stream by defining a violation line and capturing the number plate of the image if that line is crossed when the signal is red. Lane changing can also be identified similarly. These can be achieved by background subtraction or image processing techniques.
- Accident or breakdown detection: Intersections also tend to experience severe crashes due to the fact that several types of injurious crashes, such as angle and left-turn collisions, commonly occur there. Therefore, accurate and prompt detection of accidents at intersections offers tremendous benefits of saving properties and lives and minimizing congestion and delay. This can be achieved by identifying the vehicles that remain stationary for a long time in an inappropriate position such as in the middle of the road, so that parked vehicles are not included in this.
- Synchronization of traffic signals across multiple intersections: Synchronizing signals along a street can benefit the commuters as once a vehicle enters the street, it may continue with minimal stopping. [16].
- Adapting to emergency vehicles: Emergency vehicles such as an ambulance need to be given quicker passage through the traffic signals. The model can be trained to detect not just vehicles but also be able to recognize that it is an emergency vehicle and accordingly adapt the timers such that the emergency vehicle is given priority and is able to cross the signal at the earliest.

REFERENCES

[1]. S. K.Sarangi ,R.Panda & Manoranjan Dash,” Design of 1-D and 2-D recursive filters using crossover bacterial foraging and cuckoo search techniques”, Engineering Applications of Artificial Intelligence, Elsevier Science,vol.34, pp.109-121,May 2014.

[2]. Manoranjan Dash, N.D. Londhe, S. Ghosh, et al., “Hybrid Seeker Optimization Algorithm-based Accurate Image Clustering for Automatic Psoriasis Lesion Detection”, Artificial Intelligence for Healthcare (Taylor & Francis), 2022, ISBN: 9781003241409

[3]. Manoranjan Dash, Design of Finite Impulse Response Filters Using Evolutionary Techniques - An Efficient Computation, ICTACT Journal on Communication Technology, March 2020, Volume: 11, Issue: 01

[4]. Manoranjan Dash, “Modified VGG-16 model for COVID-19 chest X-ray images: optimal binary severity assessment,” International Journal of Data Mining and Bioinformatics, vol. 1, no. 1, Jan. 2025, doi: 10.1504/ijdmb.2025.10065665.

[5]. Manoranjan Dash et al.,” Effective Automated Medical Image Segmentation Using Hybrid Computational Intelligence Technique”, Blockchain and IoT Based Smart Healthcare Systems, Bentham Science Publishers, Pp. 174-182,2024

[6]. Manoranjan Dash et al.,” Detection of Psychological Stability Status Using Machine Learning Algorithms”, International Conference on Intelligent Systems and Machine Learning, Springer Nature Switzerland, Pp.44-51, 2022.

[7]. Samriya, J. K., Chakraborty, C., Sharma, A., Kumar, M., & Ramakuri, S. K. (2023). Adversarial ML-based secured cloud architecture for consumer Internet of Things of smart healthcare. IEEE Transactions on Consumer Electronics, 70(1), 2058-2065.

[8]. Ramakuri, S. K., Prasad, M., Sathiyanarayanan, M., Harika, K., Rohit, K., & Jaina, G. (2025). 6 Smart Paralysis. Smart Devices for Medical 4.0 Technologies, 112.
[9]. Kumar, R.S., Nalamachu, A., Burhan, S.W., Reddy, V.S. (2024). A Considerative Analysis of the Current Classification and Application Trends of Brain–Computer Interface. In: Kumar Jain, P., Nath Singh, Y., Gollapalli, R.P., Singh, S.P. (eds) Advances in Signal Processing and Communication Engineering. ICASPACE 2023. Lecture Notes in Electrical Engineering, vol 1157. Springer, Singapore. https://doi.org/10.1007/978-981-97-0562-7_46.
[10]. R. S. Kumar, K. K. Srinivas, A. Peddi and P. A. H. Vardhini, "Artificial Intelligence based Human Attention Detection through Brain Computer Interface for Health Care Monitoring," 2021 IEEE International Conference on Biomedical Engineering, Computer and Information Technology for Health (BECITHCON), Dhaka, Bangladesh, 2021, pp. 42-45, doi: 10.1109/BECITHCON54710.2021.9893646.
[11]. Vytla, V., Ramakuri, S. K., Peddi, A., Srinivas, K. K., & Ragav, N. N. (2021, February). Mathematical models for predicting COVID-19 pandemic: a review. In Journal of Physics: Conference Series (Vol. 1797, No. 1, p. 012009). IOP Publishing.
[12]. S. K. Ramakuri, C. Chakraborty, S. Ghosh and B. Gupta, "Performance analysis of eye-state charecterization through single electrode EEG device for medical application," 2017 Global Wireless Summit (GWS), Cape Town, South Africa, 2017, pp. 1-6, doi:10.1109/GWS.2017.8300494.
[13]. Gogu S, Sathe S (2022) autofpr: an efficient automatic approach for facial paralysis recognition using facial features. Int J Artif Intell Tools. https://doi.org/10.1142/S0218213023400055
[14]. Rao, N.K., and G. S. Reddy. “Discovery of Preliminary Centroids Using Improved K-Means Clustering Algorithm”, International Journal of Computer Science and Information Technologies, Vol. 3 (3), 2012, 4558-4561.
[15]. Gogu, S. R., & Sathe, S. R. (2024). Ensemble stacking for grading facial paralysis through statistical analysis of facial features. Traitement du Signal, 41(2), 225–240.
[16]. Chandrasekhara Reddy, T., Sirisha, G., Reddy, A.M. Smart Healthcare Analysis and Therapy for Voice Disorder using Cloud and Edge Computing, Proceedings of the 4th International Conference on Applied and Theoretical Computing and Communication Technology, iCATccT 2018, 2018, pp. 103–10б, 9001280
[17]. Chandrasekhara Reddy, T., Srivani, V., Mallikarjuna Reddy, A., Vishnu Murthy, G. Test case optimization and prioritization using improved cuckoo search and particle swarm optimization algorithm, International Journal of Engineering and Technology(UAE), 2018, 7(4.б Special Issue б), pp. 275–278
[18]. Dayaker, P., Chandrasekhara Reddy, T., Honey Diana, P., Mallikarjun Reddy, A. Recent trends in security and privacy of sensitive data in cloud computing, International Journal of Engineering and Technology(UAE), 2018, 7(4.б Special Issue б), pp. 241–245
[19]. Mallikarjuna Reddy, A., Venkata Krishna, V., Sumalatha, L. Face recognition approaches: A survey, International Journal of Engineering and Technology(UAE), 2018, 7(4.б Special Issue б), pp. 117–121
[20]. Obulesh, A., Vamshi Krishna, P., Mallikarjuna Reddy, A. Classification of lung patterns based on transition neighborhood pattern (TNP), Journal of Advanced Research in Dynamical and Control Systems, 2018, 10(15), pp. 35–41.
[21]. Daniel, G. V., Chandrasekaran, K., Meenakshi, V., & Paneer, P. (2023). Robust Graph Neural-Network-Based Encoder for Node and Edge Deep Anomaly Detection on Attributed Networks. Electronics, 12(6), 1501. https://doi.org/10.3390/electronics12061501
[22]. Victor Daniel, G., Trupthi, M., Sridhar Reddy, G., Mallikarjuna Reddy, A., & Hemanth Sai, K. (2025). AI Model Optimization Techniques. Model Optimization Methods for Efficient and Edge AI: Federated Learning Architectures, Frameworks and Applications, 87-108.
[23]. Lakshmi, M.A., Victor Daniel, G., Srinivasa Rao, D. (2019). Initial Centroids for K-Means Using Nearest Neighbors and Feature Means. In: Wang, J., Reddy, G., Prasad, V., Reddy, V. (eds) Soft Computing and Signal Processing. Advances in Intelligent Systems and Computing, vol 900. Springer, Singapore. https://doi.org/10.1007/978-981-13-3600-3_3