e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:04/April-2024 Impact Factor- 7.868 www.irjmets.com

TRAFFIC SIGNAL CONTROL SYSTEM BASED ON VEHICLE DENSITY USING YOLO DETECTION AND IMAGE PROCESSING: A PYGAME SIMULATION MODEL

Prof. Snehal Kale[*1], Swapnali Bakal[*2], Gauri Kokate[*3], Sejal Pol[*4], Anuradha Virkar[*5]

[*1,2,3,4,5]Department of Information Technology Engineering, Marathwada Mitra Mandal's College of Engineering, Pune, Maharashtra, India.

DOI : https://www.doi.org/10.56726/IRJMETS53192

ABSTRACT

Managing the increasing number of vehicles in urban areas presents a persistent challenge. Traffic narrowing disrupts daily routines, elevates stress levels, and contributes to higher carbon emissions, impacting the environment. With the rise in population, megacities need to catch up on transportation activities. An intelligent traffic management system is essential to monitor traffic density continuously and take proactive measures. Although there are designated lanes for different vehicle types, wait times at traffic signals remain unchanged. To address this, we propose leveraging artificial intelligence to gather real-time images from signals and improve the current system.

Our methodology uses the YOLO image processing technique to assess traffic density accurately. YOLO is brilliant in detecting multiple vehicles, enhancing the system's effectiveness in managing traffic jams. Intelligent monitoring technology integrates a signal-switching algorithm at intersections to optimize time distribution and mitigate traffic congestion. This approach aims to reduce vehicle waiting times, improving overall traffic flow.

Keywords: Traffic management, traffic signal optimization, YOLO, signal-switching algorithm, image processing, artificial intelligence.

I. INTRODUCTION

As cities grow, so does the number of vehicles on the road, especially in big cities. This leads to problems like traffic jams and difficulties in controlling traffic. The current traffic management systems need a lot of manpower to prevent accidents and manage long lines of vehicles at intersections. We need smarter systems and infrastructure to manage traffic better. Wireless sensor networks (WSNs) can help control traffic flow at intersections. In the past, traffic control involved manual work by traffic police using signs, lights, and whistles. Sensors and timers are now crucial for managing traffic lights.

- Manual Traffic Control: This method relies on human intervention to manage traffic flow. Traffic officers are deployed to specific areas to regulate traffic using tools such as signboards, signal lights, and whistles.
- Traditional Traffic Lights with Fixed Timers: These traffic control systems operate based on predetermined time intervals. A set time duration is programmed into the timer, causing the traffic lights to switch between red and green automatically.
- Electronic Sensor: A more advanced approach involves the installation of loop detectors or proximity sensors along the road. These sensors collect real-time data about traffic conditions, which is then used to adjust traffic signal timings accordingly.

After studying different methods for detecting vehicle density, we decided to create an adaptive traffic control system. This system recognizes objects in images and adjusts traffic signals accordingly. Traditional methods are time-consuming and rely on manual work, which is not feasible everywhere due to a shortage of manpower. We need a more efficient traffic control system that responds to real-time traffic conditions.

Challenges with Traditional Methods:

1. Limited Coverage: Using sensors like proximity detectors is costly and may not cover all areas due to budget constraints.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:04/April-2024 Impact Factor- 7.868 www.irjmets.com

2. Complex Equipment: High-quality data collection often requires expensive equipment, making it difficult to gather accurate information.

To address these challenges, we use live images from CCTV cameras at intersections to calculate real-time traffic density. We classify vehicles like cars, bikes, buses, trucks, and bicycles to adjust green signal times accurately. This helps improve traffic flow and reduce waiting times at intersections.

The YOLO (You Only Look Once) approach is employed to detect the number of cars, allowing for adjustment of traffic signal timers based on vehicle density in captured images. This optimization of green signal intervals leads to more efficient traffic flow compared to static systems, resulting in reduced delays, congestion, and waiting times. Traditional traffic signal allocation systems lack awareness of traffic density, resulting in prolonged waiting times, particularly in densely populated cities, leading to severe traffic congestion. While sensor-based traffic management partially alleviates this issue, it requires frequent maintenance. Therefore, there is a need to enhance signal time allocation to further reduce traffic congestion.

To address this, we propose an automated traffic management framework based on artificial intelligence with a comprehensive understanding of traffic density. This framework aligns with the research model proposed.

II. LITERATURE REVIEW

Reference [2] presents a traffic management system based on Arduino-UNO. The system aims to alleviate traffic congestion and reduce waiting times by utilizing a camera to capture images. These images are processed using MATLAB, where they undergo conversion to a threshold image by eliminating saturation and hues, enabling the calculation of traffic density. Connectivity between Arduino and MATLAB is established via USB and simulation packages. The Arduino adjusts the duration of the green light for each lane based on traffic count and density. However, the method has limitations. Vehicle overlap poses challenges for accurate vehicle counting, and other objects in the image, such as billboards, poles, and trees, are not distinguished from vehicles due to the black-and-white conversion process.

Qingyan Wang et al. [5] demonstrated enhancements to the YOLOv4 algorithm, improving its predictive capabilities. In their detection trial, the authors achieved a 97.58 percent area under the PR curve (AUC), surpassing the 90 percent achieved in the Vision for Intelligent Vehicles and Applications Challenge Competition. Additionally, in the recognition experiment, the mean average precision of the Improved YOLOv4 method was 2.86 percent higher than that of the original YOLOv4 technique. The Improved YOLOv4 algorithm proves to be a reliable and effective solution for real-time traffic light signal detection and recognition.

Reference [9] proposes the use of adaptive light timer control using image processing techniques and traffic density. This system consists of a microcontroller-controlled traffic light timer, high image sensing devices, MATLAB, and transmission using UART principles. However, this system fails to prioritize the authorized emergency vehicles nor to detect accidents at the intersection.

RFID (radio-frequency identification) tags utilize radio waves to transmit data about objects to an antenna/reader system. P. Manikonda, A.K. Yerrapragada, et al. [10] devised a method for detecting vehicle speed employing an RFID tag and reader. The average speed of cars detected by a specified number (N) of readers was used to calculate the average time at specific crossings. However, this approach requires continuous communication between tags and readers and mandates the installation of RFID tags in each vehicle. A. Kanungo, A. Sharma, et al. [11] adopted a hardware-free method to compute vehicle density using background separation techniques at 30 frames per second, incorporating picture matrices. The time allocated for signaling at four-way intersections was determined by vehicle density, with a range of green light duration between 10 and 60 seconds. Although the concept was promising, the study overlooked crucial aspects such as initial vehicle count and starting positions, essential for accurate analysis.

Reference [12] proposes a smart traffic light system using ANN and a fuzzy controller. This system makes use of images captured from cameras installed at traffic sites. The image is first converted to a grayscale image before further normalization. Then, segmentation is performed using the sliding window technique to count the cars irrespective of size and ANN is run through the segmented image, the output of which is used in a fuzzy

controller to set timers for red and green light using crisp output. Results had an average error of 2% with execution time of 1.5 seconds.

III. METHODOLOGY

This section outlines the methodology adopted for conducting the research, focusing on the utilization of the YOLO approach for an efficient traffic management system in regulating vehicle movements.

YOLO stands for "You Only Look Once," an advanced object detection algorithm renowned for its exceptional speed and accuracy. Operating as a sophisticated convolutional neural network (CNN), YOLO excels in recognizing multiple objects within images. It employs a clever strategy where the image is divided into regions, and a single neural network processes the entire image to predict bounding boxes and probabilities for each region. These bounding boxes are weighted based on their expected changes, effectively capturing comprehensive information contained within the image. One significant advantage of YOLO is its ability to generate a vast amount of data encompassing all the relevant details within an image. Through a single CNN, YOLO predicts various bounding boxes and class probabilities for different objects, enhancing detection performance through training on a captured set of photographs. The backbone of YOLO utilizes a CNN architecture that can be further fine-tuned to enhance processing efficiency. Darknet, an open-source neural network architecture developed in C and CUDA, serves as the foundation for YOLO. Darknet offers rapid setup and supports both CPU and GPU computing. In evaluations on ImageNet, YOLO achieves a top-1 accuracy of 72.9 percent and a top-5 accuracy of 91.2 percent when implemented with DarkNet. Darknet predominantly employs 3x3 filters for feature extraction and 1x1 filters for output channel reduction. Additionally, it utilizes global average pooling to make predictions.

A. Proposed System Overview:

Our proposed system utilizes CCTV camera images captured at traffic junctions as input for real-time traffic density calculation through image processing and object detection. The image undergoes analysis by the vehicle detection algorithm, employing YOLO. This algorithm identifies the number of vehicles belonging to various classes such as cars, bikes, buses, and trucks. The signal-switching algorithm utilizes this density, along with other relevant factors, to determine the green signal timer for each lane. Subsequently, adjustments are made to the red signal times accordingly in other lines. To prevent the starvation of any particular lane, the green signal time is constrained within maximum and minimum thresholds. Additionally, we have developed a simulation to showcase the effectiveness of the system and to compare its performance with the existing static system.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:04/April-2024 Impact Factor- 7.868 www.irjmets.com

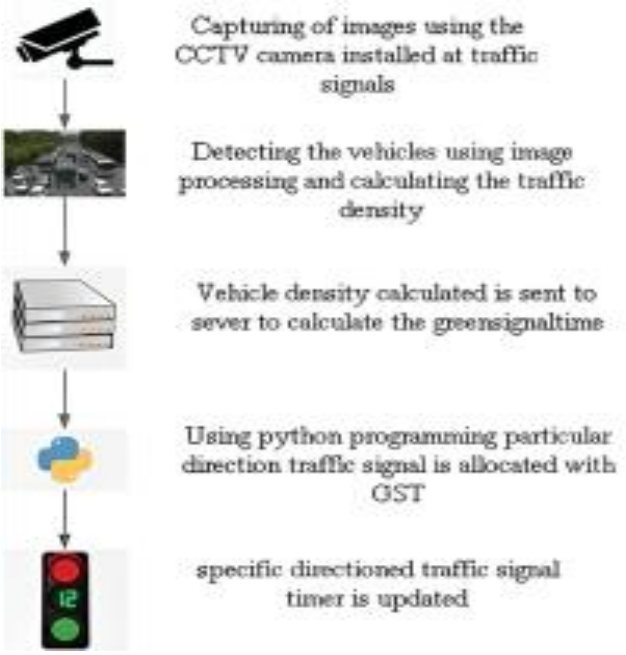

Fig 1. Proposed system

B. Vehicle Detection Module:

Our proposed system leverages YOLO (You Only Look Once) for vehicle detection, offering the desired accuracy and processing speed. A customized YOLO model was trained specifically for vehicle detection, capable of identifying various vehicle classes such as cars, bikes, buses, and trucks.

YOLO stands as an ingenious convolutional neural network (CNN) designed for real-time object detection. The algorithm operates by applying a single neural network to the entire image, subsequently dividing it into regions and predicting bounding boxes and probabilities for each region of the image. These bounding boxes are weighted according to the predicted probabilities. YOLO is renowned for its ability to achieve high accuracy while maintaining real-time performance. The algorithm "only looks once" at the image, requiring a single forward propagation pass through the neural network for predictions. Following non-maximum suppression, which ensures that each object is detected only once, YOLO outputs recognized objects along with their respective bounding boxes.

The dataset utilized for training the model was compiled by scraping images from Google and manually labeling them using LabelIMG, a graphical image annotation tool. Following data collection, the model underwent training using pre-existing weights obtained from the YOLO website. The configuration of the .cfg file used for training was adjusted to match the specifications of our model. Specifically, the number of output neurons in the final layer was set equal to the number of classes the model was intended to detect, which in our case was 4: Car, Bike, Bus/Truck, and Rickshaw. Once these configuration changes were applied, the model underwent training until the loss was significantly reduced and no further decrease was observed, indicating the completion of the training process. Subsequently, the updated weights were imported into the code and utilized for vehicle detection with the assistance of the OpenCV library. A threshold was defined to establish the minimum confidence level required for successful detection.OpenCV was then employed to draw bounding boxes around the detected objects using the labels and coordinates received from the model's output.

Figure 2 depicts test images upon which our vehicle detection model was applied. The left portion of the figure displays the original image, while the right side showcases the output obtained after applying the vehicle detection model to the image. This output includes bounding boxes surrounding the detected vehicles, accompanied by their corresponding labels.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:04/April-2024 Impact Factor- 7.868 www.irjmets.com

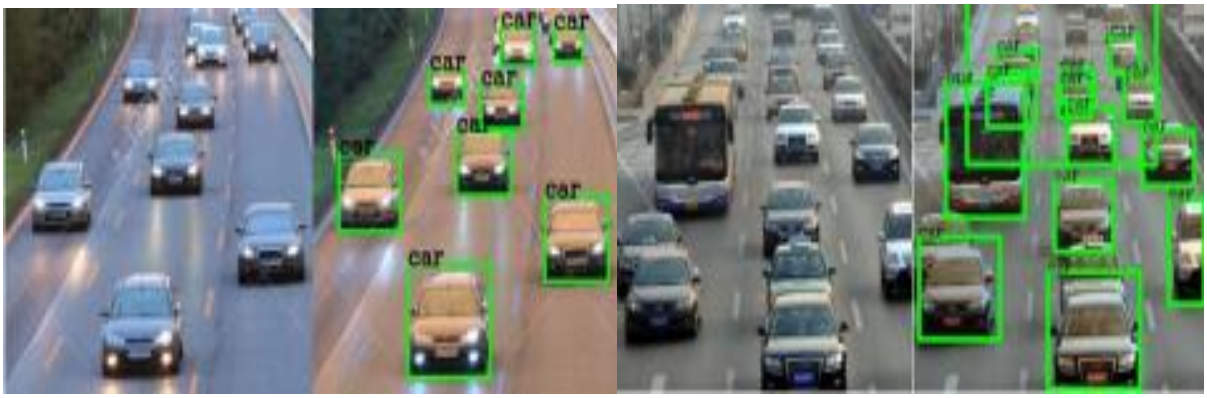

Fig. 2. Vehicle Detection Results

D. Simulation Module:

Pygame is a versatile collection of Python modules designed for crafting video games and simulations across different platforms. It encompasses computer graphics and sound libraries tailored to seamlessly integrate with Python, leveraging AI, mathematics, and Pygame itself, which extends the capabilities of the SDL library. This integration empowers developers to create comprehensive games and multimedia applications entirely in Python. Notably, Pygame is highly portable and capable of running on various platforms and operating systems. We utilize Pygame's functionalities to dynamically generate vehicles and monitor.

A simulation was developed using Pygame from scratch to replicate real-life traffic scenarios. This simulation serves as a visualization tool for the system and facilitates comparisons with the existing static system. It features a 4-way intersection equipped with four traffic signals, each displaying a timer indicating the time remaining for signal transitions (green to yellow, yellow to red, or red to green). Additionally, the number of vehicles that have crossed the intersection is displayed alongside each signal. Various types of vehicles, including cars, bikes, buses, trucks, and rickshaws, enter from all directions. To enhance realism, some vehicles in the rightmost lane are programmed to turn at the intersection, with the decision to turn being randomized during vehicle generation. Furthermore, a timer is incorporated to track the elapsed time since the start of the simulation. Figure 3 provides a snapshot of the simulation module.

Pygame is a collection of Python modules designed for developing video games across different platforms. It offers computer graphics and sound libraries tailored for use with the Python programming language, thereby extending its functionality. Pygame facilitates the creation of interactive and visually appealing games and applications.

Fig 3. Simulation module

Signal Switching Algorithm:

1. The Signal Switching Algorithm adjusts the duration of green signals based on the traffic density observed by the vehicle detection module. It also updates the timing of red signals for other lanes accordingly.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:04/April-2024 Impact Factor- 7.868 www.irjmets.com

Additionally, it cycles through the signals based on their timers, ensuring each lane gets its turn to have a green signal.

2. Input Data Format: The algorithm takes input about detected vehicles in JSON format. Each detected object has a label (type of vehicle), confidence level, and coordinates.
3. Calculating Traffic Density: The algorithm parses the input data to count the total number of vehicles of each type. Factors considered include processing time, number of lanes, vehicle count, and traffic density.
4. Setting Green Signal Time: Green signal time is determined based on traffic density and other factors. Minimum and maximum time limits are set to prevent signal starvation.
5. Adjusting Red Signal Times: The red signal times of other signals are adjusted based on the green signal time of the current signal.
6. Algorithm Operation: When the algorithm starts, it sets default times for the first signal and calculates times for subsequent signals. Separate threads handle vehicle detection for each direction. The main thread manages the timer for the current signal. When the green signal timer of the current signal reaches 0, detection threads capture the next direction's snapshot. Detected vehicles are parsed, and the next green signal timer is set. This seamless process ensures a smooth transition between signal changes without lag.
7. Image Capture and Processing: Images are captured when the time for the next signal to turn green reaches 0 seconds. The system has 5 seconds (equal to the duration of the yellow signal) to process the image, detect vehicles, and set signal times.
8. Calculating Green Signal Time: The green signal time is calculated based on the number of vehicles of each type, their average crossing time, and the number of lanes.
9. Customizing Average Crossing Time: Average crossing times for each vehicle type can be customized based on location or intersection characteristics. Data from transport authorities can be analyzed for this purpose.
10. Signal Switching Pattern: Signals switch in a cyclic pattern, similar to the current system. The order of signals and yellow signals are accounted for to maintain consistency and prevent confusion. This algorithm ensures efficient traffic management by dynamically adjusting signal times based on real-time traffic density while maintaining the familiar pattern of signal switching for drivers.

Working of the algorithm:

At the start of the algorithm, it initializes the timing for the first signal in the first cycle and calculates the timings for all subsequent signals in the first cycle, as well as for all signals in the following cycles. It operates with two threads: one for detecting vehicles in each direction and another for managing the timer of the current signal. When the timer for the current green signal reaches 0 seconds, or when the red signal timer for the next green signal ends, the vehicle detection threads capture the next direction's snapshot. Then, the results are processed, and the timer for the next green signal is set. This process runs in the background while the main thread continues counting down the timer for the current green signal. This seamless coordination ensures smooth timing transitions without delays.

Images are captured when the timer for the next signal to turn green reaches 0 seconds, providing a 5-second window (equivalent to the duration of the yellow signal) to process the image. During this time, the algorithm detects the number of vehicles in the image, calculates the green signal time based on the vehicle count and their characteristics (such as average speeds and acceleration times), and adjusts the timings for the current signal and the red signal for the next signal accordingly. This allows for efficient traffic management and ensures that each class of vehicles has sufficient time to cross the intersection.

IV. RESULTS AND ANALYSIS

This section presents the outcomes and analyses derived from the development and examination of a robust traffic light management system leveraging the YOLO algorithm. Visual aids such as photographs, graphs, and tables are included to enhance the clarity and understanding of the undertaken work.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:04/April-2024 Impact Factor- 7.868 www.irjmets.com

A) Evaluation of Vehicle Detection Module:

The vehicle detection system was tested using different test images that included varying numbers of vehicles. The detection accuracy ranged from 70% to 80%, which is acceptable but not ideal. The main reason for the moderate accuracy is the lack of quality of the dataset. To enhance the system's performance, training the model with real-world footage from traffic cameras could lead to better accuracy.

B) Evaluation of the proposed adaptive system:

To evaluate how the new adaptive system performs against the current static system, 10 simulations were conducted for each system, for 5 minutes each. The simulations used different traffic patterns across four directions. The performance was assessed based on the number of vehicles passing through the intersection per unit of time. Additionally, the idle time of the signal (when it is green but no cars are passing through) was compared. This affects how long vehicles wait and the length of queues at the other signals.

The results were organized in a table, showing the number of vehicles that passed through each lane, as well as the total number of vehicles that passed overall.

Table 1. Simulation Results of Current System

N0.	L1	L2	L3	L4	Total
1	85	109	44	50	287
2	121	58	49	29	257
3	94	50	60	58	262
4	85	48	71	59	263
5	97	29	100	34	260
6	64	53	80	47	244
7	56	108	19	78	261
8	87	68	70	33	258
9	52	75	101	25	253
10	66	82	40	48	236

Table 2. Simulation Result of Proposed System

No.	L1	L2	L3	L4	Total
1	70	60	72	60	252
2	112	72	30	32	246
3	80	91	30	72	273

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:04/April-2024 Impact Factor- 7.868 www.irjmets.com

4	74	82	40	63	259
5	30	55	47	33	165
6	23	30	52	67	172
7	69	72	72	80	293
8	33	40	52	92	217
9	121	49	48	31	249
10	100	10	8	4	122

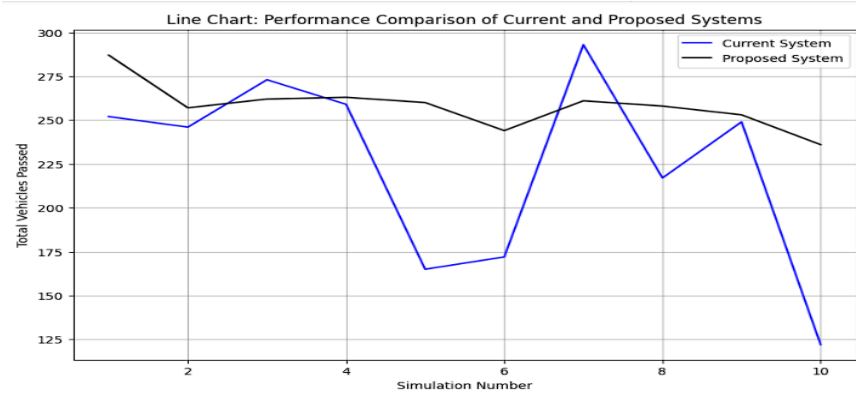

Fig. 4. Comparison of the current static system and proposed adaptive system

As shown in Figure 4, the new adaptive system consistently outperforms the current static system, regardless of how traffic is distributed. The performance improvement varies based on the degree of traffic unevenness across lanes. Greater differences in traffic distribution lead to greater performance enhancements.

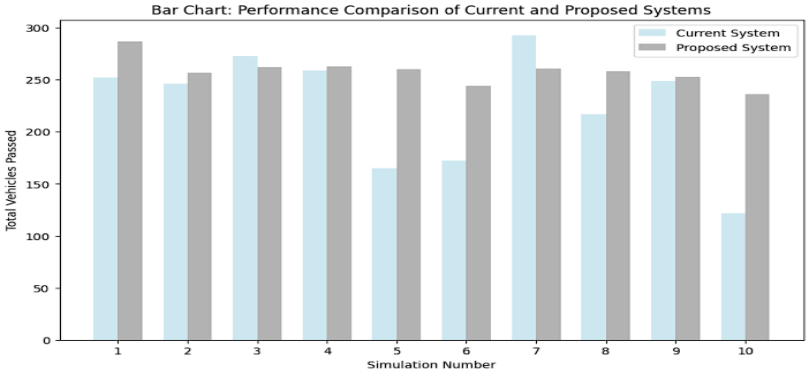

Fig. 5. Performance Comparison of Current and Proposed System

V. CONCLUSION AND FUTURE WORK

In conclusion, the proposed system dynamically adjusts the green signal duration based on traffic density at the signal, prioritizing the direction with heavier traffic for longer green signal periods compared to less congested directions. This adaptive approach aims to minimize delays, alleviate congestion, and reduce waiting times, leading to decreased fuel consumption and pollution levels. Simulation results indicate a notable 23% enhancement in the number of vehicles passing through the intersection compared to the current system, signifying a significant improvement in traffic flow efficiency. Further refinement through training the model

with real-life CCTV data holds the potential to enhance the system's performance even further. Furthermore, the proposed system offers distinct advantages over conventional intelligent traffic control systems like Pressure Mats and Infrared Sensors. Its deployment incurs minimal costs since it utilizes existing CCTV camera footage from traffic signals, eliminating the need for additional hardware in most cases. Any necessary adjustments typically involve minor alignment rather than significant infrastructure investments. Additionally, maintenance costs are significantly reduced compared to alternatives like pressure mats, which are prone to wear and tear due to constant exposure to road pressure. By leveraging existing CCTV infrastructure, the proposed system presents an economical and sustainable solution for traffic monitoring and management. Its seamless integration with major city CCTV networks holds the potential to significantly enhance traffic management efficiency.

The project can be expanded to incorporate the following functionalities aimed at optimizing traffic management and reducing congestion:

1. Accident or breakdown detection: Intersections are prone to severe crashes, including angle and left-turn collisions, which can result in significant property damage and injuries. Thus, the timely and precise detection of accidents at intersections is crucial for saving lives, preserving property, and reducing congestion and delays.
2. Responding to emergency vehicles: Emergency vehicles like ambulances require expedited passage through traffic signals. The system can be trained not only to detect vehicles but also to recognize emergency vehicles. Subsequently, it can adjust signal timers to prioritize the passage of emergency vehicles, enabling them to cross the signal promptly.
3. Synchronization of traffic signals across multiple intersections: Coordinating signals along a street can enhance the commuting experience by minimizing the need for vehicles to stop frequently once they enter the street.

VI. 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), Mysore, 2017, pp. 99-103, doi: 10.1109/CTCEEC.2017.8454966.

[2] A. Vogel, I. Oremović, R. Šimić and E. Ivanjko, "Improving Traffic Light Control by Means of Fuzzy Logic," 2018 International Symposium ELMAR, Zadar, 2018, pp. 51-56, doi: 10.23919/ELMAR.2018.8534692.

[3] Adarsh P., Rathi, P., Kumar, M. (2020). "YOLO v3-Tiny: Object detection and recognition using one stage improved model." 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS). doi:10.1109/ icaccs48705.2020.9074315.

[4] Pranav Shinde, Srinand Yadav, Shivani Rudrake, Pravin Kumbhar. "Smart Traffic Control System using YOLO." 2019 IEEE 8th Data-Driven Control and Learning Systems Conference (DDCLS). doi:10.1109/ddcls.2019.8908873.

[5] Wang Q, Zhang Q, Liang X, Wang Y, Zhou C, Mikulovich VI. "Traffic Lights Detection and Recognition Method Based on the Improved YOLOv4 Algorithm." Sensors, vol. 22, no. 1, 2022, 200. https://doi.org/10.3390/s22010200.

[6] Tai Huu - Phuong Tran, Jae Wook Jeon. "Accurate Real-Time Traffic Light Detection Using YOLOv4." 2020. DOI: 10.1109/IC CE-Asia49877.2020.9277063

[7] Alharbi, A., Halikias, G., Sen, A.A.A., et al. "A framework for dynamic smart traffic light management system." Int. J. Inf. Technol. vol. 13, 2021, 1769–1776. https://doi.org/10.1007/s41870- 021-00755-2

[8] 'Traffic Signal Synchronization'. [Online]. Available: https://www.cityofirvine.org/signal-operations-maintenance/traffic signal-synchronization.

[9] Siddharth Srivastava, Subhadeep Chakraborty, Raj Kamal, Rahil, Minocha, "Adaptive traffic light timer controller", IIT KANPUR, NERD MAGAZINE.

[10] Manikonda P, Yerrapragada AK, Annasamudram SS. (2011). "Intelligent traffic management system." 2011 IEEE Conference on Sustainable Utilization and Development in Engineering and Technology (STUDENT). doi:10.1109/stu dent.2011.6089337.

[11] Kanungo A, Sharma A, Singla C. (2014). "Smart traffic lights switching and traffic density calculation using video processing." 2014 Recent Advances in Engineering and Computational Sciences (RAECS). doi:10.1109/raecs.2014.6799542.

[12] Renjith Soman "Traffic Light Control and Violation Detection Using Image Processing"." IOSR Journal of Engineering (IOSRJEN), vol. 08, no. 4, 2018, pp. 23-27.

[13] A. A. Zaid, Y. Suhweil and M. A. Yaman, "Smart controlling for traffic light time," 2017 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), Aqaba, 2017, pp. 1-5, doi: 10.1109/AEECT.2017.8257768.

[14] Pranav Shinde, Srinand Yadav, Shivani Rudrake, Pravin Kumbhar. "Smart Traffic Control System using YOLO." 2019 IEEE 8th Data-Driven Control and Learning Systems Conference (DDCLS). doi:10.1109/ddcls.2019.8908873.

[15] Rizwan P., Suresh K.,Babu MR. "Real-time smart traffic management system for smart cities by using the Internet of Things and big data." 2016, International Conference on Emerging Technological Trends (ICETT). doi:10.1109/ icett.2016.7873660.