FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 13, Issue 2 | Pages 287-295| e-ISSN: 2347-470X

Real-Time Traffic Light Optimization Using Yolov9 and Length-Based Metrics

Dr. Nilesh B. Korade[1*], Dr. Mahendra B. Salunke[2], Dr. Amol A. Bhosle[3], Dr. Sunil M. Sangve[4], Dhanashri M. Joshi[5], Gayatri G. Asalkar[6], and Swati R. Paralkar[7]

[1]*Department of Computer Science and Engineering (Artificial Intelligence), Vishwakarma Institute of Technology, Pune, India; nilesh.korade.ml@gmail.com*
[2]*Department of Computer Engineering, PCET's, Pimpri Chinchwad College of Engineering and Research, Pune, India; mahendra.salunke@pccoer.in*
[3]*Department of Computer Science and Engineering, MIT Art, Design and Technology University, Pune, India, amolabhosle@gmail.com; amolabhosle@gmail.com*
[4]*Department of Artificial intelligence and Data science, Vishwakarma Institute of Technology, Pune, India; sunil.sangve@vit.edu*
[5]*Department of Computer Engineering, JSPM's Rajarshi Shahu College of Engineering, Pune, India; jdhanashrim@gmail.com*
[6]*Department of Computer Science and Engineering (Data Science), Vishwakarma Institute of Technology, Pune, India; gayatri.teke@gmail.com*
[7]*Department of Computer Engineering, JSPM's Rajarshi Shahu College of Engineering, Pune, India; swatianantwar@gmail.com*

***Correspondence:** *Nilesh B. Korade, nilesh.korade.ml@gmail.com;*

ABSTRACT- The Indian traffic control system faces lots of difficulties due to the increasing volume of vehicles, ineffective systems for traffic administration during peak hours, and the frequent need for manual intervention due to the inadequate performance of traffic signals in managing heavy traffic flow. Traditional traffic lights in India have defined timings for each lane, which frequently cause longer traffic jams in lanes with more traffic. This study presents an intelligent traffic control system that incorporates the YOLOv9 model for real-time traffic length prediction and intelligently allocates green, red, and orange signal timings. YOLOv9 builds a bounding box that allows it to compute vehicle density precisely by enclosing the initial and final cars in every frame. As a comparison with traditional fixed-time methods, the proposed approach recalculates traffic signal timings at each cycle based on the recorded length of traffic. In order to ensure that emergency services respond more quickly, the system efficiently prioritizes emergency vehicles by quickly moving them from their lane when they are spotted. This adaptive strategy aligns signal duration with real traffic demand, boosting overall traffic efficiency and regulating traffic flow by reducing unnecessary wait times for low-traffic lanes. In comparison to fixed-timing systems and object detection strategies, our research on adaptive traffic systems reveals a 66% to 77% decrease in vehicle delay. Compared to traditional fixed-timing approaches, the proposed method demonstrates significant enhancements in effectively managing traffic congestion.

Keywords: Traffic Signal, Optimization, YOLOv9, Bounding Box, Object Detection.

ARTICLE INFORMATION
Author(s): Dr. Nilesh B. Korade, Dr. Mahendra B. Salunke, Dr. Amol A. Bhosle, Dr. Sunil M. Sangve, Dhanashri M. Joshi, Gayatri G. Asalkar, and Swati R. Paralkar;
Received: 29/01/2025; **Accepted**: 12/05/2025; **Published**: 18/06/2025;
E- ISSN: 2347-470X;
Paper Id: IJEER 2901-01;
Citation: 10.37391/ijeer.130212
Webpage-link:
https://ijeer.forexjournal.co.in/archive/volume-13/ijeer-130212.html

Publisher's Note: FOREX Publication stays neutral with regard to jurisdictional claims in Published maps and institutional affiliations.

1. INTRODUCTION

India has a population of 1.44 billion, with over 354 million registered vehicles and an average of 20 million new vehicle registrations each year. The major cities in India, such as Mumbai and Kolkata, are frequently congested by traffic due to their extreme density, where more than 24,000 people are living per square kilometer [1]. In the morning, a substantial part of the population travels to work, and in the evening, they return to their homes, creating peak traffic periods. On continuous weekends or major festivals like Diwali and Dussehra, many people travel to their native place, leading to specific lanes experiencing prolonged congestion at particular times, making traffic problems worse [2]. Fixed-timing signals waste important signal time and lead to congestion as they are incapable of adjusting the flow of traffic in real time, which can result in a long vehicle queue at some intersections while other lanes remain empty [3]. Approximately twenty-four thousand individuals in India lose their lives every day as a consequence of traffic-related delays in receiving medical attention. Twenty percent of the deaths of emergency patients are caused by traffic congestion. Manual traffic signals are frequently deactivated in situations of extreme traffic congestion, leading to the active management and direction of vehicle movement by police officials on the site in order to alleviate the congestion [4]. The fixed-interval traffic signal-based traffic management system has the following difficulties:

• The allocation of identical time to each lane leads to inefficiencies as lanes with heavier traffic remain uncleared within the allocated time, whereas lanes with lighter traffic remain idle, leading to a considerable waste of valuable traffic time.
• Ineffective traffic clearing frequently causes emergency patients to not reach the hospital in time, which can result in preventable deaths [5].
• To deal with a lengthy traffic situation, traffic lights must be turned off, requiring personal intervention.
• Extended traffic clearance times increase vehicle emissions significantly, leading to higher carbon emissions, adversely impacting the environment [6].

Irrespective of the actual traffic density, traditional traffic control systems usually use predetermined signal timings, which can result in inefficiencies such as longer wait times, more traffic, and higher fuel consumption. According to several studies, the dynamic modification of signal durations in response to actual traffic demand in adaptive, real-time systems might greatly improve traffic flow, reduce carbon emissions, and increase road safety [7]. Some studies track the number of vehicles in each lane to adjust the timing of traffic signals. When traffic density is high, this strategy faces restrictions because object identification algorithms frequently have trouble correctly identifying cars in lengthy lines. Only seeing car roofs limits the algorithm's field of vision, which results in insufficient detection and erroneous corrections for long traffic lines [8].

Our proposed approach presents a revised application of length-based bounding box metrics when combined with YOLOv9 for real-time signal optimization, designed specifically for Indian traffic situations, which are frequently characterized by non-uniform vehicle shapes, unstructured lanes, and significant occlusion. In contrast to prior studies, which rely exclusively on vehicle count or fixed-length assumptions, our approach computes the actual physical span of traffic by employing bounding boxes from the first to the last identified vehicle in the lane. We implemented an automated traffic control system that uses YOLOv9 to represent the traffic density in each lane by drawing bounding boxes around vehicles, and then the length of traffic is precisely estimated in real time by analyzing these bounding boxes. The proposed system automatically adjusts the green, orange, and red signal timings for optimal traffic flow in real-time based on the estimated traffic length for each lane. The detection of bounding boxes for each lane, calculation of traffic length, and adjustment of traffic signals are performed in real-time during each traffic light cycle. Additionally, the system uses YOLOv9 to recognize emergency vehicles, such as ambulances, and the system prioritizes lane clearance as soon as an emergency vehicle is detected until the vehicle has passed [9].

After being cleared, the signals resume their regular sequence until the cycle completes. In contrast to conventional YOLO object prediction techniques, which frequently fail to identify cars at larger distances because of visual constraints, the suggested solution uses bounding boxes to determine the traffic length effectively. By removing the need for vehicle detection, this method improves accuracy, especially in crowded or distant areas. By converting bounding box lengths from pixels to meters, the system enables real-time traffic length estimation. This guarantees accurate lane-by-lane traffic light timing adjustments based on real traffic density, improving traffic control and reducing wait times. Without requiring considerable object detection model retraining or adaptation, the technique is easily scalable to multi-lane intersections and different traffic circumstances. Because of this, it can be used in a variety of urban situations. Perfect vehicle detection is a crucial aspect of YOLO object prediction, but it might be hampered by occlusions, bad weather, or poor image quality. The suggested approach lessens mistakes brought on by these restrictions by concentrating on traffic length estimation rather than individual car counts.

2. LITERATURE REVIEW

The author P. Kunekar et al. implemented a convolutional neural network (CNN) and a self-adaptive and robust optimizer for deep learning-based congestion reduction in their smart transportation management system. Using the algorithm, the traffic density is determined, considering the quantity of vehicles and their kind, such as trucks, cars, or bicycles. The lack of designated lanes for various vehicle categories makes it difficult to determine the exact number of vehicles. Furthermore, it might be challenging to get an accurate vehicle count when larger vehicles are positioned in front, which can obscure smaller vehicles behind them [10].

Vehicle tracking in smart traffic management is particularly difficult when there are obstructions, clutter, changes in the lighting in the real world, scene circumstances, and camera viewpoint. The A. A. Ahmed et al. proposed an urban traffic control system that incorporates techniques such as vehicle counts, process control, and lane evaluation. Through a line-of-sight link, the infrared transmitter used in the invention continuously sends signals to the ground-based detector. Two timers are triggered when a vehicle interrupts the initial infrared sensor, marking entry at time 't' and exit at 't+1', allowing for the estimation of the vehicle's length. Speed is calculated as the interval between two infrared connection interruptions spaced one meter apart. The controller modifies the lights according to the computed speed and traffic density. The results showed that the average waiting time for all vehicles during the simulation period was reduced by 25.98%, and for no interference movement flow, it was reduced by 34.16% [11].

The author, M. Sharma et al., counts the number of vehicles crossing a predetermined line while detecting and tracking vehicles on a video stream using YOLO and SORT. The reference line is positioned 100–120 meters ahead of the red light in order to determine the number of cars that are crossing that line. The system optimized the traffic lights by assigning them time according to the traffic behavior in real-time, depending on the calculated count of vehicles for each side of the traffic light. The system's weaknesses are that the accuracy gets impacted as dark vehicles may not always fulfill the detection criterion and when vehicles are close to one another or have significant shadows. That night, sceneries are difficult to resolve since headlight beams can produce large areas that meet threshold criteria [12].

To address the issue of traffic congestion, S. M. M. Azad et al. implemented intelligent traffic signal control using a reinforcement learning (RL) approach. This approach introduced a smart traffic light system designed for both a single intersection and two interconnected intersections, aiming to minimize vehicle waiting times at intersections. When compared to a traditional static traffic signal system, the suggested solution showed a 34% decrease in average vehicle queue time for the single intersection scenario. The performance of independent and interconnected agents was investigated in the context of two intersections. According to the results, overall waiting periods were shorter when the agents were treated as independent than when they were integrated. Agents were able to make well-informed decisions by using data that was taken from discrete action regions [13].

K. Balint et al. introduced an innovative solution to the adaptive traffic signal control (TSC) problem by integrating a unique combination of state representation, a novel reward mechanism, and a simplified action space. In order to provide clear recommendations for decision-making, the suggested framework makes use of the policy gradient algorithm, which forecasts the likelihood of choosing among the available options for every state. The training environment for the algorithm is designed using SUMO (Simulation of Urban Mobility), a widely used simulation tool for urban traffic systems. After training, the model's performance was examined in a variety of novel settings to thoroughly assess how well it could generalize. Promising developments in adaptive traffic management are demonstrated by the combination of improved state representation, a new reward formulation, and a simplified action space [14].

The increasing number of vehicles on roads has highlighted the need for advanced and efficient traffic signal control systems. To address this issue, the study proposed by A. R. Mou leverages image-based traffic density estimation using cameras mounted on traffic posts. To detect and quantify vehicles, this simple and efficient method is utilized, which integrates a YOLO classifier with a fuzzy controller. This system periodically captures images, processes them with a cascade classifier for calibration, and utilizes the fuzzy controller to evaluate the density of traffic. This method evaluates vehicle density across multiple images captures to dynamically modify traffic signal timings. The output function of the fuzzy controller makes it possible to make decisions about signal operation in real time, which makes the traffic management system more responsive. These advancements demonstrate a promising direction for modern traffic signal optimization [15]. Even with dedicated lanes for different kinds of vehicles, passenger waiting periods at traffic signals are still somewhat lengthy. Novel approaches that make use of artificial intelligence have been investigated to address these issues by B.R. Prathap et al. In order to optimize time allocation at intersections, this intelligent monitoring system combines dynamic signal-switching algorithms with image processing techniques. To determine the density of traffic, YOLOv4 was integrated to evaluate the captured real-time pictures at traffic intersections. According to the simulation results, the suggested solution performs significantly better than the current approach in terms of the volume of traffic that crosses the intersection [16].

N. Sakhare et al. introduced a novel approach leveraging the Internet of Things (IoT) and advanced image processing techniques utilizing the YOLO algorithm. This adaptive traffic control system precisely identifies and counts vehicles by analyzing real-time data from camera-monitored lanes. A dynamic algorithm uses this data to determine the best waiting durations for every lane, allowing for an effective distribution of signal timings. By significantly reducing average vehicle wait times, this strategy improves traffic flow overall [17].

This study tries to resolve the outlined research gap by exposing the limitations of existing approaches. While several studies have employed YOLO-based object identification to estimate traffic density, the majority rely on vehicle counting, line-crossing logic, or image-based classification, which are susceptible to inaccuracies in occluded, overlapped, or shadowed situations. In addition, traditional regulation of traffic methods, including some adaptive systems, fails to dynamically modify signal timings in real time depending on spatial parameters such as traffic length. YOLO-based algorithms frequently struggle to detect objects effectively in circumstances with dense, extremely congested, unstructured, or partially obscured traffic, reducing the reliability of vehicle count-based traffic control systems.

3. METHODOLOGY

The proposed traffic management system captures images of traffic in each lane and utilizes the YOLOv9 object detection model to identify and highlight regions of traffic by drawing bounding boxes [18-20]. These bounding boxes' lengths, expressed in pixels, are subsequently translated into actual units of measurement such as meters. In order to maximize traffic flow, the system dynamically allots signal durations for each lane based on the overall traffic length determined for all lanes. In situations where an emergency vehicle is detected, the system prioritizes clearing the lane containing the emergency vehicle, maintaining the green signal until the vehicle has passed.

YOLOv9 outperforms its predecessors, YOLOv8 and YOLOv7, in terms of handling obstructed and congested scenes due to the incorporation of GELAN and PGI modules. These architectural advancements boost detection robustness, making YOLOv9 suitable for dynamic and visually complicated traffic scenarios. In this study, we have used YOLOv9 as the primary object detection model and fine-tuned it using a custom-built dataset designed specifically for Indian traffic circumstances. The collection was selected from video footage captured at several metropolitan junctions at different times of day, yielding over 3,000 annotated frames representing a variety of vehicle

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 13, Issue 2 | Pages 287-295| e-ISSN: 2347-470X

types, including cars, buses, etc., under tough conditions such as occlusion, clutter, and changing lighting. The model was fine-tuned for 100 epochs with this dataset. The training configuration used a batch size of 16, an image resolution of 640×640, and used the SGD optimizer with momentum. Bounding box regression was performed using a 0.001 learning rate and a cosine decay scheduler, and Complete IoU (CIoU) loss was applied.

Traffic lane recordings were captured and processed with YOLOv9 to create bounding boxes around vehicles. The entire traffic length has been estimated using these detections in order to dynamically change signal timings. The visualization was carried out in PyCharm to show traffic accumulation and time allocation, ensuring smooth vehicle flow when the green signal is enabled. The architecture of the proposed system is illustrated in *figure 1*.

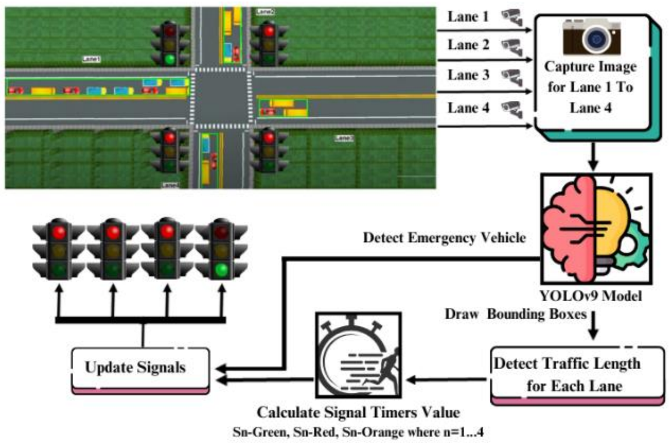

Figure 1. Architecture of proposed traffic control system

3.1. Draw Bounding Boxes

The computer vision architecture YOLOv9, which offers object detection and image segmentation, introduces innovative ideas such as Generalized Efficient Layer Aggregation Network (GELAN) and Programmable Gradient Information (PGI) to improve the accuracy and efficiency of object detection tasks. With attention on preventing information loss and improving network efficiency, the YOLOv9 framework offers an innovative approach to solving fundamental challenges in object detection using deep learning [21].

3.1.1. The Information Bottleneck Principle

The loss of information during data transformation in a neural network is described by the Information Bottleneck Principle [22].

$$I(X, X) \geq I(X, f_\theta(X)) \geq I(X, g_\phi(f_\theta(X))) \qquad (1)$$

Data X loses important information required for accurate forecasts as it moves through the layers (f_θ and g_ϕ) of a deep neural network, leading to model convergence being hampered and gradients becoming unstable. The f and g stand for transformation functions with parameters θ and ϕ, respectively; I stands for mutual information.

3.1.2. Reversible Functions

YOLOv9's reversible functions tackle the issue of information loss in deep networks' forward passes, which is crucial for applications requiring accurate object detection. These functions guarantee that the original input data can be accurately recreated from the network's outputs by enabling the reversal of data transformations [23].

$$X = v_\zeta\left(r_\psi(X)\right) \qquad (2)$$

The ψ and ζ are the parameters for the forward and reverse transformation functions, r and v, respectively.

3.1.3. Programmable Gradient Information

PGI offers programmable control over gradient computation to emphasize certain learning process aspects, including preserving important information from the input or intermediate layers, minimizing gradient dilution, particularly in very deep networks, and improving feature learning for particular tasks like localization or object detection. Let ∇L denote the gradient of the loss function L with respect to the model's parameters [24]. PGI introduces a scaling factor α at *layer l*, such that:

$$\nabla L_1 = \alpha_1 . \nabla L_1^{raw} \qquad (3)$$

Where, ∇L_1^{raw} is the raw gradient at layer l.

3.1.4. Generalized Efficient Layer Aggregation Network

An improved network architecture called the GELAN was created to maximize the aggregation of features from several layers in deep learning models. It is especially useful for object detection tasks performed by YOLOv9 as it aims to retain computational efficiency, increase gradient propagation, and improve information flow across the network [25].

3.2. Traffic Length Calculation

In order to determine the actual length of traffic, we installed a camera 10 feet above the ground to capture pictures of different traffic lengths. Bounding boxes were drawn around the traffic areas in the captured photos using YOLOv9. Subsequently, the bounding box lengths in pixels were noted along with the corresponding actual traffic lengths in meters. By plotting pixel lengths against real-world distances in meters, we derived the following equation to estimate the relationship between pixel lengths and actual distance. where y is the length of the traffic in meters, x is the length of the bounding box in pixels, and α and n are curve-fitted constants [26-28].

$$y = \alpha * x^n \qquad (4)$$

3.3. Calculate Signal Timers

The proposed system allocates signal time to each lane based on the traffic length, ensuring that each lane has sufficient time to clear its vehicles efficiently.

T_l^n=Traffic length for each lane where n is number of lanes.

S_t^n=Time allocated for each lane where n is number of lanes.

T_l=Total traffic length where $T_l = T_l^1 + T_l^2 + .. + T_l^n$

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 13, Issue 2 | Pages 287-295 | e-ISSN: 2347-470X

S_t=Total traffic signal time where $S_t = S_t^1 + S_t^2 + .. + S_t^n$

The traffic clearance time for each lane is determined using *equation 5*.

$$S_t^n = \frac{T_l^n * S_t}{T_l} \quad (5)$$

Consider a scenario with four lanes, each controlled by red, green, and orange traffic signals. Following the calculation of signal durations, the system will activate the green light for *Lane 1*, followed by the orange and red signals in sequence. For *lanes 2* and *3*, the initial signal is red, transitioning to green and then orange before returning to red. In the case of *Lane 4*, the cycle begins with a red light, followed by green and orange, with subsequent recalculation performed at the end of each cycle to determine signal timings for the next cycle. The calculation for the *lane 1* signal is expressed below. In this study, we assume a fixed duration S_0 for all orange signals, which are triggered immediately after the green light. As a warning indication, the orange signal alerts vehicle drivers to be ready for the coming red light and proceed with care.

$$S_{ot}^n = S_o \quad (6)$$

Lane1 signals (Green [S_{gt}^1], Orange [S_{ot}^1], Red [S_{rt}^1]) calculations are as below:

$$S_{gt}^1 = S_t^1 - S_o \quad (7)$$

$$S_{rt}^1 = S_t - S_t^1 \quad (8)$$

Lane2 signals (Red [$S_{r_1t}^2$], Green [S_{gt}^2], Orange [S_{ot}^2], Red [$S_{r_2t}^2$]) calculations are as below:

$$S_{r_1t}^2 = S_t^1 \quad (9)$$

$$S_{gt}^2 = S_t^2 - S_o \quad (10)$$

$$S_{r_2t}^2 = S_t - (S_t^1 + S_t^2) \quad (11)$$

Lane3 signals (Red [$S_{r_1t}^3$], Green [S_{gt}^3], Orange [S_{ot}^3], Red [$S_{r_2t}^3$]) calculations are as below:

$$S_{r_1t}^3 = S_t^1 + S_t^2 \quad (12)$$

$$S_{gt}^3 = S_t^3 - S_o \quad (13)$$

$$S_{r_2t}^3 = S_t - (S_t^1 + S_t^2 + S_t^3) \quad (14)$$

Lane4 signals (Red [[S_{rt}^4]], Green [S_{gt}^4], Orange [S_{ot}^4]) calculations are as below:

$$S_{rt}^4 = S_t^1 + S_t^2 + S_t^3 \quad (15)$$

$$S_{gt}^4 = S_t^4 - S_o \quad (16)$$

3.4. Update Signal

The Update Signal module adjusts the traffic signals based on the calculated timings. When an emergency vehicle exists, it overrides the normal sequence, halts other signals, and prioritizes clearing traffic in the lane of the emergency vehicle. After that, the system continues to function in accordance with the algorithm's instructions.

3.5. Operational Flow

The algorithms for the Dynamic Traffic Signal Control and the Emergency Vehicle Monitoring are described in detail below.

3.5.1. Dynamic Traffic Signal Control Algorithm

```
Input: Real-time images from lane cameras
Output: Optimized signal durations for each lane (Red, Green, Orange)
Begin
  while True do
    // Step 1: Capture pictures for every lanes.
    For each lane n from 1 to N do
      image[n] ← captureImageFromCamera(n)
    // Step 2: Detect vehicles with YOLOv9 to estimate traffic length.
    For each lane n from 1 to N do
      boundingBoxes[n] ← YOLOv9(image[n])
      pixelLength[n] ← measureTrafficPixelLength(boundingBoxes[n])
    // Step 3: Transform pixel length into actual meter scale.
    For each lane n from 1 to N do
      trafficLength[n] ← pixelToMeter(pixelLength[n])
    // Step 4: Determine the total traffic length.
  totalTrafficLength ← Σ (trafficLength[n] for n = 1 to N)
    // Step 5: Define total signal cycle time (St)
    totalSignalTime ← predefinedCycleTime
    // Step 6: Estimate signal time for each lane by examining traffic ratios.
    For each lane n from 1 to N do
      signalTime[n] ← (trafficLength[n] * totalSignalTime) / totalTrafficLength
    // Step 7: Allocate green, orange, and red signal timings.
    For each lane n from 1 to N do
      orangeTime[n] ← So // fixed
      greenTime[n] ← signalTime[n] - So
      If n = 1 then
        redTime[n] ← totalSignalTime - signalTime[n]
      Else if n < N then
        redTime1[n] ← Σ(signalTime[k]) for k = 1 to n−1
        redTime2[n] ← totalSignalTime - Σ(signalTime[k]) for k = 1 to n
      Else
        redTime[n] ← Σ(signalTime[k]) for k = 1 to n−1
    // Step 8: Activate signals as per estimated times.
```

```
        updateTrafficSignals(greenTime,orangeTime,
redTime)
        // Step 9: Wait for the signal cycle to end
        Wait(totalSignalTime)
    end while
End
```

3.5.2. Emergency Vehicle Monitoring Algorithm

Input: Real-time images from lane cameras

Output: Adaptive traffic signal behavior to prioritize emergency vehicles.

```
Begin
    while True do
        // Step 1: Capture pictures for every lane.
        For each lane n from 1 to N, do
            image[n] ← captureImageFromCamera(n)
        // Step 2: Recognize emergency vehicles with YOLOv9
        For each lane n from 1 to N, do
            emergencyDetected[n] ← DetectEmergencyVehicleYOLOv9(image[n])
        // Step 3: Check for the appearance of emergency vehicles.
        If ∀n emergencyDetected[n] == False, then
            executeStandardTrafficAlgorithm()
            continue loop
        End If
        // Step 4: Execute emergency vehicle logic
        For each lane n from 1 to N do
            If emergencyDetected[n] == True, then
                If isGreenSignalActive(n) then
                    maintainGreenSignal(n)
                    While emergencyDetected[n] == True, do
                        image[n] ← captureImageFromCamera(n)
                        emergencyDetected[n] ← DetectEmergencyVehicleYOLOv9(image[n])
                    End While
                    resumeNormalTrafficControl()
                Else
                    activateGreenSignal(n)
                    maintainGreenSignal(n)
                    While emergencyDetected[n] == True, do
                        image[n] ← captureImageFromCamera(n)
                        emergencyDetected[n] ← DetectEmergencyVehicleYOLOv9(image[n])
                    End While
                    resumeNormalTrafficControl()
                End If
            End If
        End For
        // Step 5: Restart monitoring loop
    end while
End
```

4. RESULTS AND DISCUSSION

4.1. Assessment Metrics

Intersection Over Union (IoU): The IoU metric evaluates how well the anticipated bounding box aligns with the ground truth bounding box. Higher IoU indicates better localization.

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} = \frac{B_p \cap B_g}{B_p \cup B_g} \quad (17)$$

The predicted bounding box is represented by B_p, while the ground truth bounding box is represented by Bg. The $B_p \cap B_g$ is the intersection or overlapping area, and $B_p \cup B_g$ is the union of the two boxes.

Mean Average Precision (mAP): The mAP measures the object identification model's precision across various thresholds. It determines how well your model identifies and classifies items.

$$\text{mAP} = \frac{1}{N}\sum_{i=1}^{N} AP_i \quad Where\ \text{Precision} = \frac{TP}{TP + FP} \quad (18)$$

where N is the total number of classes and AP_i is the average precision for class *i*.

Traffic Length Estimation (TLE) Error (%): It predicts traffic length (based on object count or bounding box span) and the actual length of traffic on the road.

$$TLE\ Error(\%) = \left\| \frac{L_{pred} - L_{true}}{L_{true}} \right\| * 100 \quad (19)$$

L_{pred} refers to the predicted length, and L_{true} refers to the actual traffic length measured manually or using accurate tools.

Adaptive Signal Accuracy: How accurately the system adjusts signal duration in response to actual traffic length.

$$Adaptive\ Signal\ Accuracy = 1 - \frac{T_{adaptive} - T_{ideal}}{T_{ideal}} \quad (20)$$

Where $T_{adaptive}$ refers to the signal time calculated by your system and T_{ideal} refers to the manually or historically optimal signal time.

Computation Time per Frame: The average time, measured in milliseconds, that is required to process a single frame, which includes object recognition, bounding box drawing, and traffic length calculation.

$$T_{frame\ (ms)} = T_{end} - T_{start} * 1000 \quad (21)$$

Emergency Vehicle Detection Accuracy: How well emergency vehicles such as ambulances or fire engines are spotted.

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 13, Issue 2 | Pages 287-295 | e-ISSN: 2347-470X

$$\text{Emergency Vehicle Detection Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (22)$$

TP refers to the correctly detected emergency vehicles, TN refers to the correctly detected non-emergency vehicles, FP refers to the non-emergency vehicle falsely marked as emergency, and FN refers to the emergency vehicle falsely marked as non-emergency.

Handling Occlusions: A qualitative parameter that represents how effectively the model identifies partially obscured cars can be quantified by measuring recall for counting the missed detections in occluded zones.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (23)$$

4.2. Comparison of Object Detection and the Bounding Box Approach

The weaknesses of conventional object detection-based traffic management are demonstrated by a comparison between object detection alone and the suggested bounding-box-based traffic length prediction in *figures 2* and *3*.

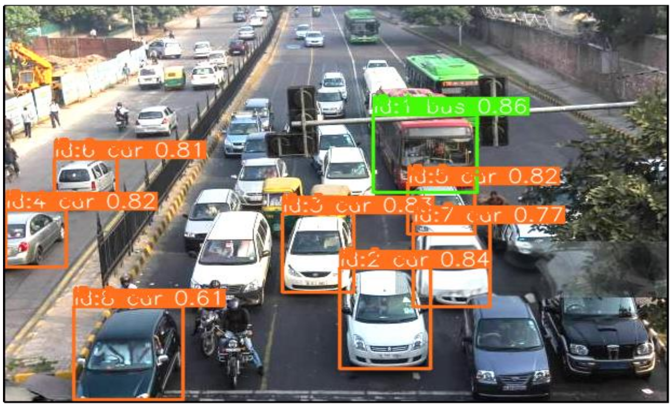

Figure 2. YOLOv9 Object Detection Method to Calculate Traffic Length

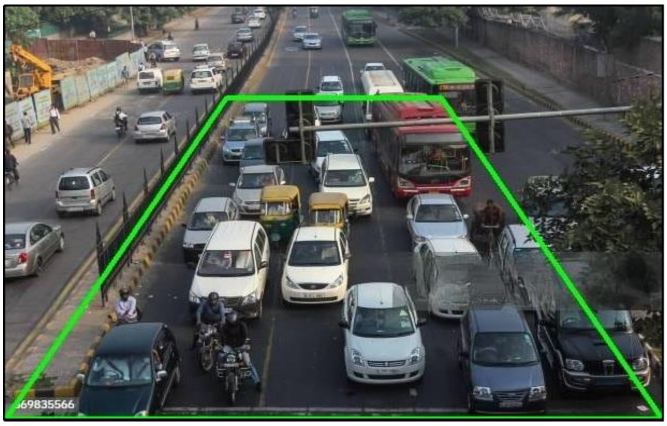

Figure 3. YOLOv9 Bounding Box Approach to Calculate Traffic Length

Even though object detection is effective for identifying individual vehicles, it often fails to detect all objects within a captured image because of occlusions, the surrounding environment, or object size variations. As a result, relying solely on object detection leads to inaccurate traffic estimations and inadequate signal modifications. On the other hand, the bounding box-based traffic length prediction enables accurate signal timing calculations for dynamic modifications and offers a more thorough perspective of the traffic volume. *Table 1* presents an examination of YOLOv9 object detection and bounding box-based length estimation using assessment metrics. The performance in handling occlusions is shown in *table 2*.

Table 1. YOLOv9 Object Detection *vs* Bounding Box-Based Length Estimation

Parameter	YOLOv9 (Vehicle Counting)	YOLOv9 (Bounding Box-Based)
Mean Average Precision (mAP)	0.77	0.89
Intersection Over Union (IoU)	0.78	0.91
Traffic Length Estimation Error (%)	22%	4%
Computation Time per Frame (ms)	38ms	40ms
Emergency Vehicle Detection Accuracy	74%	91%
Adaptive Signal Accuracy	85%	94%

Table 2. Performance in Handling Occlusions

Parameter	YOLOv9 (Vehicle Counting)	YOLOv9 (Bounding Box-Based)
Precision under Occlusion (%)	76.3%	89.5%
Recall under Occlusion (%)	72.1%	87.8%
F1-Score under Occlusion (%)	74.1%	88.6%

4.2. Evaluation of Bounding Box-Based Traffic Length Estimation through Simulation

To visualize the proposed system, PyCharm was utilized, where traffic is randomly generated across four lanes. Each vehicle is given a certain speed at which it will cross the intersection during the green light. The visualization of YOLOv9's bounding box detections and the dynamic adjustments of signal timings were effectively rendered through its built-in graphical capabilities. The simulation results for traffic light optimization are shown in *figure 4*, where each lane's length is estimated using bounding boxes. This process ensures an accurate measurement of traffic density in each lane. *Figure 5* demonstrates the subsequent adjustment of traffic signals based on the calculated lane lengths. The average delay per vehicle has been estimated under heavy and moderate traffic conditions by adjusting the vehicle arrival speeds in each lane. These visual representations confirm the proposed system's potential to effectively monitor and adapt to traffic conditions, optimizing signal timings to reduce congestion.

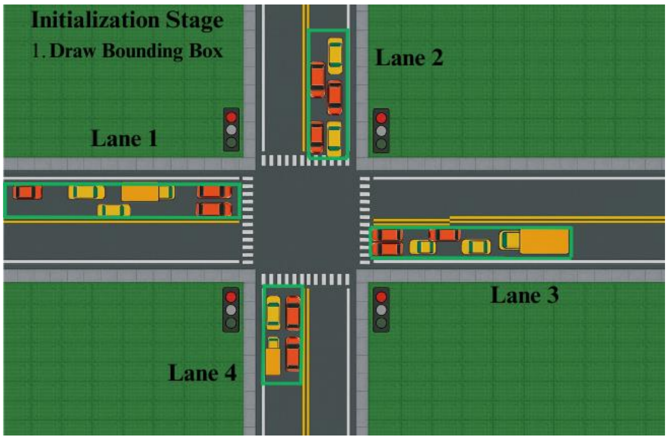

Figure 4. Traffic Signal Optimization Simulation (Draw Bounding Box) Using PyCharm

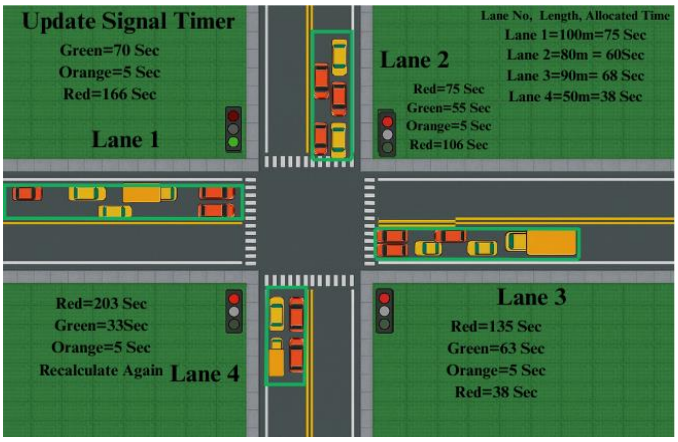

Figure 5. Traffic Signal Optimization Simulation (Calculate Signal Time) Using PyCharm

Experimental results demonstrate that the bounding box-based approach allows for more precise signal timing calculations, ensuring smoother traffic flow and reducing congestion. The technology automatically adjusts to changing traffic patterns, demonstrating its ability to manage complex urban traffic situations. Furthermore, the system's ability to prioritize emergency vehicles ensures timely clearance of such lanes, adding a layer of safety and efficiency. *Table 3* provides a comparative analysis of the fixed-time signal method, object detection-based method, and bounding box-based approach in terms of traffic light optimization.

Table 3. Comparison of Fixed-Time Signal, Fixed-Time Signal, and Bounding-Box-Based Method

Parameter	Fixed-Time Signal	Object Detection Method [29-31]	Bounding-Box-Based Method
Traffic Signal	Static	Based on the	Based on lane-
Traffic Density	Low	Moderate	High
Adaptability	None	Moderate	High
Average Delay per	30 to 90	20–60 seconds	10 to 30 seconds
Environmental	High	Moderate	Low
Performance in	Poor	Moderate	Excellent

These results demonstrate how well the suggested methodology handles the difficulties with modern traffic management. The average delay in a bounding-box-based approach is 10–30 seconds per vehicle, compared to 20–60 seconds for object detection and 30–90 seconds for fixed-time signals [30, 31]. With dynamic signal duration adjustments based on traffic length, the suggested approach minimizes vehicle delays by roughly 66% to 77%. This translates to an estimated time saving of 20 to 60 seconds per vehicle.

5. CONCLUSIONS

The proposed traffic light optimization system demonstrates a significant improvement over traditional fixed-timing methods and conventional object detection-based approaches. The approach offers a more precise and dynamic method of controlling traffic lights by estimating traffic length using YOLOv9. The proposed approach relies on bounding box-based traffic length estimation, in contrast to vehicle count-based approaches, which frequently fail to identify vehicles at longer distances due to picture quality or occlusion. Signal timings for each lane can be precisely calculated and adjusted by converting this length from pixels to meters. This approach guarantees a more effective flow of traffic, reduces delays, and improves flexibility in response to actual traffic situations, especially in variable conditions. The results highlight the potential of combining deep learning-based traffic length detection with dynamic signal adjustment to address the limitations of existing techniques. This technology enables smarter and more responsive urban infrastructure by establishing the groundwork for future developments in intelligent traffic management systems. Our study on adaptive traffic systems suggests a 15% to 30% reduction in vehicle delay compared to fixed-timing systems. The proposed traffic light optimization system opens several avenues for future research and development. To further improve the accuracy of traffic signal modifications, one important avenue is the incorporation of other environmental elements, including weather, road infrastructure, and pedestrian movements. To provide a more inclusive traffic management framework, the system can also be expanded to handle multimodal traffic, such as bicycles and public transportation.

REFERENCES

[1] Geetam, T.; Rahul, G.; Kavi, B. Road Safety in India: Status Report 2023. T*ransportation Research & Injury Prevention Centre Indian Institute of Technology Delhi* 2023. pp. 1-70.

[2] Archana, V.; Anuj, S.; Vanajakshi, L. A Naturalistic Data-Driven Study of Indian Traffic Conditions and Driver's Physiological Stress. *Transportation in Developing Economies* 2022, vol. 8, no. 23, doi: 10.1007/s40890-022-00158-3.

[3] Avinash, P. R.; Ashwini R. C.; Chaitanya, K,; Prashant P.; Shanshank, K. S. Deep Learning based smart traffic light system using Image Processing with YOLO v7. *4th International Conference on Circuits, Control, Communication and Computing (I4C) Bangalore India* 2022, pp. 129-132, doi: 10.1109/I4C57141.2022.10057696.

[4] Dungar, S.; Pritikana, D.; Indrajit, G. Driver behavior modeling at uncontrolled intersections under Indian traffic conditions. *Innovative Infrastructure Solutions* 2024, vol. 9, no. 124, doi: 10.1007/s41062-024-01425-5.

[5] Ayam, B.; Piyush, S.; Anupam, K.; Usha, C. IoT-Based Vehicle Class Detection for Smart Traffic Control. *Innovations in Computational Intelligence and Computer Vision* 2024, pp. 161-170, doi: 10.1007/978-981-97-6992-6_13.

[6] Qinghui, Z.; Diyi, Z.; Haoshi, L.; Yuping, H. KCS-YOLO: An Improved Algorithm for Traffic Light Detection under Low Visibility Conditions. *Machines*, vol. 12, no. 8:557, doi:10.3390/machines12080557.

[7] Moses, O.; Opeyemi, O.; Omotayo, O. An Effective Traffic Management Framework for Congestion Prediction and Re-Routing Using Hybridized Techniques. *International Journal of Intelligent Transportation Systems Research* 2024, vol. 22, pp. 687–706, doi: 10.1007/s13177-024-00425-0.

[8] Mohammed, A. A. A.; Aaqif, A.A.; Hong, F.; Rehab, A. I.; Saeed, H. A.; Ammar, H. An improved YOLO-based road traffic monitoring system. *Computing* 2021, vol. 103, pp. 211–230, doi: 10.1007/s00607-020-00869-8.

[9] Tausif, D.; Anirudh. G.; Jitendra, V. T. Object detection using YOLO: challenges, architectural successors, datasets and applications. *Multimedia Tools and Applications* 2023, vol. 82, pp. 9243–9275, doi: 10.1007/s11042-022-13644-y.

[10] Pankaj, K.; Yogita, N.; Richa, M.; Shantanu, M.; Eshan, M.; Eshan, M.; Yashashri, M. Traffic Management System Using YOLO Algorithm. Engineering Proceedings 2023, vol. 59, no. 1, doi:10.3390/engproc2023059210.

[11] Ahmad, A. A. A.; Khulood, A. M.; Shadi, A.; Eman, A. M. Smart Traffic Scheduling for Crowded Cities Road Networks. *Egyptian Informatics Journal* 2022, vol. 23, pp. 163-176, doi: 10.1016/j.eij.2022.10.002 1110-8665.

[12] Moolchand, S.; Ananya, B.; Vaibhav, K.; Pramod, G.; Tariq, H. S. Intelligent Traffic Light Control System Based on Traffic Environment Using Deep Learning. *IOP Conf. Series: Materials Science and Engineering* 2022, vol. 1022, doi: 10.1088/1757-899X/1022/1/012122.

[13] Seyedeh, M. M. A.; Ramazani A. Smart control of traffic lights based on traffic density in the multi-intersection network by using Q learning. *Discover Artifcial Intelligence* 2023, vol. 3, no. 39, doi: 10.1007/s44163-023-00087-z.

[14] Kővári, B.; Tettamanti, T.; Bécsi, T. Deep Reinforcement Learning based approach for Traffic Signal Control. *Transportation Research Procedia* 2022, vol. 62, pp. 278-285, doi: 10.1016/j.trpro.2022.02.035.

[15] Afsana, R. M.; Mariofanna, M.; Sumaiya, S. P. Intelligent Traffic Control System using YOLO Algorithm for Traffic Congested Cities. *2023 Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE)* 2023, doi: 10.1109/CSCE60160.2023.00331.

[16] Boppuru, R. P.; Kukatlapalli, P. K.; Cherukuri, R. C.; Javid, H. AI-Based Yolo V4 Intelligent Traffic Light Control System. Journal of Automation, *Mobile Robotics and Intelligent Systems* 2022, vol. 16, no.4, pp. 53-61, doi: 10.14313/JAMRIS/4-2022/33.

[17] Nitin, S.; Mrunal, H.; Gokul, B.; Omkar, M.; Trupti, S.; Anup, I. Smart Traffic: Integrating Machine Learning, and YOLO for Adaptive Traffic Management System. *International Journal of Intelligent Systems and Applications in Engineering* 2024, vol. 12, no. 12s, pp. 347–355.

[18] Peiyuan, J.; Daji, E.; Fangyao, L.; Ying, C.; Bo, M. *Procedia Computer Science* 2022, vol. 199, pp. 1066-1073, doi: 10.1016/j.procs.2022.01.135.

[19] Uddagiri, S.; Phani, P. S.; Parvathaneni, N. S.; Akash Kumar Bhoi, Paolo, B.; Akash, K. B. Statistical Analysis of Design Aspects of Various YOLO-Based Deep Learning Models for Object Detection. *International Journal of Computational Intelligence Systems* 2023, vol. 16, no. 126, doi: 10.1007/s44196-023-00302-w.

[19] Mehmet, Ş. G.; Gültekin, I. A new YOLO-based method for real-time crowd detection from video and performance analysis of YOLO models. *Journal of Real-Time Image Processing* 2023, vol. 20, no. 5, doi: 10.1007/s11554-023-01276-w.

[20] Li, J.; Feng, Y.; Shao, Y.; Liu, F. IDP-YOLOV9: Improvement of Object Detection Model in Severe Weather Scenarios from Drone Perspective. *Appl. Sci.* 2024, *14*, 5277. doi:10.3390/app14125277.

[21] Huang, G., Huang, Y., Li, H.; Guan, z.; Li, X.; Zhang, G.; Li, W.; Zheng, X. An Improved YOLOv9 and Its Applications for Detecting Flexible Circuit Boards Connectors. *Int J Comput Intell Syst* 2024, vol. 17, no.261. doi: 10.1007/s44196-024-00669-4.

[22] An, R.; Zhang, X.; Sun, M.; Wang, G. GC-YOLOv9: Innovative smart city traffic monitoring solution. *Alexandria Engineering Journal* 2024, vol. 106, pp. 277-287, doi: 10.1016/j.aej.2024.07.004.

[23] Wang, C. Y.; Yeh, I-H.; Liao, H. Y. M. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. Computer Vision – ECCV 2024, *Lecture Notes in Computer Science* 2024, vol. 15089, pp. 1–21, doi: 10.1007/978-3-031-72751-1_1.

[24] Song, L.; Wu Y.; Zhang, W. Research on Fire Detection Based on the Yolov9 Algorithm. *2024 IEEE 4th International Conference on Electronic Technology, Communication and Information (ICETCI), Changchun, China* 2024, pp. 1398-1406, doi: 10.1109/ICETCI61221.2024.10594570.

[25] Korade, N. B.; Zuber, M. Forecasting Stock Price Using Time-Series Analysis and Deep Learning Techniques. *Data Engineering and Applications: Proceedings of the International Conference, IDEA 2K22* 2024, vol.1, doi: 10.1007/978-981-97-0037-0_31.

[26] Korade, N. B.; Salunke, M. B.; Bhosle, A. A.; Kumbharkar, P. B.; Asalkar, G. G.; Khedkar, R. G.; Strengthening Sentence Similarity Identification through OpenAI Embeddings and Deep Learning. *International Journal of Advanced Computer Science and Applications* 2024, vol. 15, no.4, pp. 821-829, doi: 10.14569/IJACSA.2024.0150485.

[27] Rakesh, S.; Hegde, N. P.; A New Method to Control Traffic Congestion by Calculating Traffic Density. *International Journal of Intelligent Systems and Applications in Engineering* 2022, vol. 10, no. 1s, pp. 01–08.

[28] Boppuru, P. R.; Kukatlapalli, P. K.; Chowdary, C. R.; Hussain, J. AI-based YOLO V4 Intelligent Traffic Light Control System. *Journal of Automation, Mobile Robotics and Intelligent Systems* 2023, vol .16, no. 4, pp. 53-61, doi: 10.14313/JAMRIS/4-2022/33.

[29] Kamble, S. J.; Kounte, M. R.; Application of improved you only look once model in road traffic monitoring system. *International Journal of Electrical and Computer Engineering* 2022, vol. 13, no. 4, pp. 4612-4622, doi: 10.11591/ijece.v13i4.pp4612-4622.

[30] Biswas, S.; Acharjee, S.; Ali, A.; Chaudhari, S. S. YOLOv8 based Traffic Signal Detection in Indian Road. 2023 7th *International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech)* 2023, pp. 1-6, doi: 10.1109/IEMENTech60402.2023.10423520.

[31] Sravanthi, J.; Revathi, P.; Kumar, B. Y.; Chandra, P. S. J. Traffic Monitoring System for Signal Duration Control based on YOLOv8. *2024 International Conference on Computational Intelligence for Green and Sustainable Technologies (ICCIGST)* 2024, pp. 1-9, doi: 10.1109/ICCIGST60741.2024.10717619.