IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Traitement du Signal
Vol. 43, No. 2, April, 2026, pp. 799-810
Journal homepage: http://iieta.org/journals/ts

Toward Relieving Traffic Congestion: A Smart Traffic Light Network Using Computer Vision

Sahar Hakami[1*], Shatha Al-Hasani[2], Ghada Al-Zahrani[2], Sara Al-Withinani[2], Nebras Al-Shareef[2], Afnan Aldhahri[3], Huda Hakami[4]

[1] STC Solutions, Riyadh 11372, Saudi Arabia
[2] Department of Computer Science and Artificial Intelligence, College of Computing, Umm Al-Qura University, Makkah 24382, Saudi Arabia
[3] Department of Software Engineering, College of Computing, Umm Al-Qura University, Makkah 24382, Saudi Arabia
[4] Department of Computer Science, College of Computers and Information Technology, Taif University, Taif 21944, Saudi Arabia

Corresponding Author Email: shakami@solutions.com.sa

https://doi.org/10.18280/ts.430220

Received: 8 January 2026
Revised: 6 April 2026
Accepted: 17 April 2026
Available online: 30 April 2026

Keywords:
traffic control, smart traffic management, computer vision, machine learning, You Only Look Once

ABSTRACT

Congestion is one of the most pressing challenges in the transportation sector. Beyond contributing to driver stress and travel delays, traffic congestion significantly increases fuel consumption and air pollution. Over the years, congestion has worsened in most cities worldwide, and conventional solutions have proven insufficient in mitigating the problem. To address this issue, a smart traffic light network solution is proposed in this paper. The proposed approach employs image processing and machine learning techniques to estimate traffic density using real-time images from junction cameras. A mathematical model is then applied to optimize the timing of green and red signals and to facilitate dynamic switching between them, considering the traffic density across all approaches of an intersection. Experimental results for a fine-tuned YOLOv5 vehicle detection model demonstrate high performance in detecting vehicles of five main types, namely, car, motorcycle, truck, bus, and bicycle. According to a simulation model implemented in the Pygame environment, the proposed green signal timing method shows a significant increase in the number of vehicles passing through the intersection compared to static signal timing and single-road density–based timing approaches. These results validate the effectiveness of the proposed method in managing traffic congestion, reducing travel time, and supporting the advancement of smart city infrastructures.

1. INTRODUCTION

At present, many leading countries are competing to develop innovative solutions for building smarter urban environments. Within this context, the transportation sector has received considerable attention, with traffic congestion recognized as one of its most critical challenges. Traffic congestion is a significant issue in urban transportation [1] and is commonly described as a condition characterized by reduced travel speeds, increased travel times, and long vehicle queues. However, several features have been proposed as indicators of congestion [2], and consequently, the term "traffic congestion" lacks a universally accepted definition [3].

As the increase in vehicle traffic has not been accompanied by an increase in road capacity, this results in many congestion points, which lead to a rise in accidents, affect economic growth, and increase greenhouse gas emissions. Traffic congestion leads to waste fuels, causes air pollution, and increases stress and frustration among motorists and passengers [4]. Furthermore, accidents tend to rise as traffic volume and congestion levels rise [5].

In most urban areas worldwide, traffic congestion has progressively worsened over the years. On average, drivers in the United States waste approximately 375 liters of gasoline annually because of traffic congestion, while delays caused by intersections account for 12–55% of daily commuting time [6]. According to the TomTom Traffic Index of 2021 [7], Riyadh experienced a slight increase in congestion levels between 2020 and 2021, rising from 22% to 23%. At a congestion level of 23%, a 30-minute trip is extended by an additional 7 minutes compared to free-flow conditions.

The primary mechanism for regulating traffic movement worldwide is the traffic light system. Traditionally, traffic lights operate sequentially according to a preprogrammed schedule, which is effective in scenarios where vehicle density remains relatively constant. However, in practice, vehicle density typically fluctuates across different directions. Consequently, to address traffic congestion more effectively, the implementation of a smarter traffic light system is required [8].

The goal of smart traffic light management is to increase the efficiency of traffic signal control systems by utilizing improved tools, techniques, and equipment. Recently, a number of studies aimed at developing adaptive traffic light systems to foster intelligent transportation systems. In a dynamic traffic light system, the signals are adjusted dynamically and automatically in response to varying traffic conditions. For example, Gandhi et al. [9] proposed an algorithm that estimates green signal duration based on the road density in the intersection. Recently, the study of Mahato [10] enhanced the fixed-time strategy of traffic light signals by designing a reinforcement learning (RL) environment with multiple agents across different intersections. In the Polatlı Sanayi Intersection, the researchers of Yalçınlı et al. [11] proposed to replace the fixed-time traffic signaling system with an adaptive traffic signal control model that dynamically adjusts green signal duration using real-time feedback from a fusion of Bluetooth sensors and inductive loop detectors. The proposed adaptive model causes 20.24% reduction in delay times according to the Simulation of Urban Mobility (SUMO) traffic simulator.

This research proposes an innovative solution to tackle the problem of traffic congestion by introducing a smart adaptive traffic light system leveraging AI and computer vision. The key activity of the proposed system is to intelligently change traffic light signals based on street conditions such as total vehicles and their type in the whole intersection at a given moment, and the road condition in terms of congestion, taking into consideration other lanes in the intersection. The system uses computer vision and machine learning to determine the green signal duration for the traffic light network. The contributions of this paper are as follows:

- Fine-tune YOLOv5 and YOLOv7 on the vehicle dataset to enhance the detection of vehicles at the intersection.
- Propose a mathematical equation, based on the number and the type of detected vehicle, that computes the green signal duration considering the situation of the entire roads in the intersection.
- Evaluate the effectiveness of the proposed equation of green signal duration estimation using the Pygame simulation library.

The rest of the paper is structured as follows: Section 2 reviews previous work. In Section 3, we present our methodology. Sections 4 and 5 explain the experimental setup and experimental results, respectively. Lastly, the paper is concluded by a discussion and conclusion in Sections 6 and 7.

2. RELATED WORK

With the growing number of vehicles globally in recent years, the issue of traffic congestion has garnered significant research attention. Current advancements in traffic light network recognition systems can be broadly categorized into two main approaches, namely, hardware-based systems and software-based systems. The following sections review the two approaches in detail.

2.1 Hardware-based systems

A substantial body of research addresses the problem of traffic congestion through hardware-based systems leveraging embedded platforms, sensor networks, and programmable logic to enable real-time, low-latency decision making at intersections. In the study of Firdous and Niranjan [12], the authors proposed a smart traffic management system that uses an Arduino Uno with an ATmega328 microcontroller to automatically adjust signal timing based on traffic density. Traffic conditions are measured using digital infrared (IR) sensors, which detect vehicles by analyzing reflected signals. These roadside sensors regulate traffic density by modifying traffic signal timings as needed. The system employs LED-based traffic signals, with two LEDs assigned to each lane. To develop a functional prototype of a smart traffic signal capable of automatically adapting its timing to traffic direction, the authors also utilized solar panels as a renewable power source.

In the same context of hardware-oriented traffic management solutions, Alaidi et al. [13] proposed a smart traffic light system that dynamically controls signal timing based on the real-time number of vehicles at each approach of an intersection. The system was implemented using an Arduino, a camera, and infrared sensors. Infrared sensors were installed on all four roads at a distance of 30 meters from the intersection to detect vehicle presence. Based on the detected traffic volume, the green signal is granted when the number of vehicles on a given approach reaches a predefined threshold. The proposed approach balances traffic flow across all directions and achieves signal synchronization at four-way intersections. Experimental results showed an improvement in average waiting time at each intersection, reducing it to approximately five to six minutes.

Atta et al. [14] investigated the application of Radio-Frequency Identification (RFID) technology to alleviate traffic congestion and detect blockages at street intersections by using RFID readers and labels as sensing elements. The main objective is to transform the traditionally fixed timing of traffic signals into a dynamic system that adjusts in real time according to current traffic conditions. In their design, IoT-enabled sensors are used to monitor traffic density and regulate the dynamic timing of traffic signals to reduce congestion. A notable limitation of this system is that it requires every vehicle to be equipped with a passive RFID tag in order to be detected and tracked by the readers.

Abishek et al. [15] proposed a wireless sensor network (WSN)-based approach to address traffic congestion by making signal timing responsive to real-time traffic conditions. The system deploys wireless sensors along the approaches to an intersection to detect traffic parameters such as vehicle count, which are communicated to a local controller that dynamically adjusts both the sequence and duration of green signals to accommodate changing traffic flow. By adapting signal timing based on sensor feedback rather than fixed schedules, the system aims to maximize the number of vehicles that pass through the intersection and reduce average waiting times. Simulation results indicated that the proposed model can increase the number of vehicles passing through an intersection by approximately 7% compared to traditional fixed-time approaches.

Salama et al. [16] presented a design of an integrated intelligent system for managing and controlling traffic lights with the help of photoelectric sensors. The FPGA processor was used in this model. One of the most important criteria in this system is the installation and implementation of sensors because the traffic lights are scheduled and controlled according to the sensor values based on an algorithm with proper relative weight for each of the directions. In addition, the system can be programmed for emergency scenarios, such as passing the president, ministers, ambulances, and fire engines, through a technology based on active radio frequency detection.

Zhao et al. [17] introduced a design of an intelligent traffic control system based on DSP and NIOS II. The proposed system utilizes dual-CPU and logic control in an FPGA (Field Programmable Gate Array), which involves functions like remote control and cross-phase adjustment. Recently, integrated systems combining an FPGA with microcontroller platforms such as Arduino have been investigated to provide rapid sensor data acquisition and actuator control for real-time signal timing decisions and improved intersection safety [18]. Compared to hardware-based traffic signal controllers that depend on specialized sensing and processing units, the proposed vision-based software approach enables rapid deployment, reduced hardware complexity, and seamless scalability using existing surveillance systems.

2.2 Software-based systems

Another research direction focuses on addressing traffic congestion through software-based solutions. The authors proposed an expert system called the Traffic Lights Expert System (TLES) [19]. To determine the appropriate dynamic cycle time at the intersections, TLES employs rule-based knowledge representation combined with evidential reasoning as the inference engine. However, the main drawbacks of this approach lie in the limitations of TLES itself, particularly the challenges of knowledge acquisition as well as the high costs of maintenance and development.

The authors introduced a calendar-based traffic congestion management system that utilizes a history-based traffic management algorithm to anticipate traffic flow on congested city streets by leveraging year-round historical data [20]. The core idea is to determine the green/red times for each direction on a busy intersection using the recorded traffic history data. Furthermore, the authors propose a robust heuristic for predicting future traffic loads on streets leading to signal-controlled intersections using historical traffic patterns.

By integrating IoT with image and video processing capabilities, Razavi et al. [21] introduced a novel approach to traffic light control. The authors implemented two scheduling models: the first is based on vehicle density, while the second is based on vehicle count. In the first model, images are processed on a Raspberry Pi by extracting edges from the original images, converting them into white pixels on a black background, and then calculating the overlap percentage between the instantaneous traffic image and a reference image. This information is subsequently fed into the scheduling algorithm to dynamically determine the green light duration for each direction; however, this model does not account for interactions with other directions at the intersection. The second model uses live video for video processing, so that the number of vehicles passing through the main streets leading to the crossroads is determined. Therefore, the location of the camera to determine the number of vehicles is far from the crossroad, and on the path leading to the crossroad, and video processing is performed using the video background removal method.

In the study of Gandhi et al. [9], the suggested solution employs image processing techniques to estimate traffic density using real-time images captured by cameras at intersections. An algorithm is then applied to adjust traffic light phases according to the detected vehicle density. However, the main limitation of this system is that the green light duration for a given lane is determined independently of traffic conditions in the other lanes.

For Vehicular Ad Hoc Networks (VANETs), Khekare et al. [22] proposed a smart city framework consisting of Intelligent Traffic Lights (ITLs) that transmit traffic data and warning messages. The goal of this framework is to enable onboard units to make informed travel decisions and avoid congested roadways.

Badura and Lieskovsky [23] proposed an intelligent traffic system model that uses existing cameras at road intersections to perform traffic monitoring. The system relies on image processing techniques, particularly foreground and background modeling, to analyze traffic conditions. The processed data are then transmitted through a mobile ad hoc network, enabling flexible and real-time data delivery without dependence on fixed infrastructure.

Kanungo et al. [24] proposed a smart traffic light switching system based on video processing using a MATLAB simulator. Traffic congestion is estimated by calculating the spaces occupied by vehicles and the empty spaces along each traffic path, which reduces the image matrix and enables congestion measurement for all red-light directions. Based on these measurements, the green signal is assigned to the direction with the highest congestion level. Once a path has been given the green signal, its congestion value is reset to zero, while congestion levels in the remaining red-light directions remain unchanged. In addition, the green signal duration is determined according to the number of vehicles that can pass per second for each congestion level. Simulation results showed a 35% improvement compared to conventional traffic light systems, in which equal green signal durations are assigned to all directions.

Recent research has increasingly applied RL and deep reinforcement learning (DRL) techniques to adaptive traffic signal control problems due to their ability to learn dynamic control policies directly from traffic state information without explicit modeling of traffic dynamics. A study by Bouktif et al. [25] has shown that hybrid DRL models can simultaneously decide both signal phases and phase durations to improve control flexibility, demonstrating performance gains in simulation environments compared to traditional methods. Federated deep RL approaches further enhance model generalization and learning efficiency across different local traffic domains while preserving data privacy [26]. The survey conducted by Saadi et al. [27] summarized a range of RL-based strategies, highlighting the use of discrete and continuous action spaces, reward function engineering, and advanced policy learning techniques.

Despite that, RL and multi-agent systems have demonstrated strong performance in traffic signal control; these methods often require extensive training data, high computational resources, and complex reward engineering, which can limit their real-time deployment and scalability. In contrast, the proposed approach relies on interpretable mathematical models and real-time vehicle detection, enabling efficient deployment without the need for prolonged training phases or specialized hardware.

3. METHODOLOGY

The following section presents a detailed overview of the methodology followed in this study to develop the proposed smart traffic light network system, with the goal of managing traffic congestion. The methodology encompasses: study design and procedure, data acquisition and preparation, and testing and evaluation methods. Each of these steps has been tailored to address the challenges and objectives identified in the initial stages of the study.

3.1 Study design and procedure

This study evaluated the effectiveness of integrating You Only Look Once (YOLO) object detection models with a signal scheduling algorithm to develop a smart traffic light network system. As shown in Figure 1, the study employs a comprehensive methodology integrating three main components to address traffic management challenges. First, the detection model utilizes deep learning architectures, YOLOv5 and YOLOv7, trained on a custom dataset explained in the next section, to accurately identify vehicles in various traffic scenarios. This dataset comprised images of various vehicles captured under diverse environmental conditions to simulate real-world scenarios. Second, data from the detection model feeds into the scheduling model, which calculates optimal green light durations based on real-time intersection congestion levels. The scheduling algorithm's timing system was developed using a modified approach in the study of Gandhi et al. [9], based on the queue length method and density ratio method (explained in subsections 3.1 and 3.2). Finally, a simulation model was adopted and used to evaluate system performance.

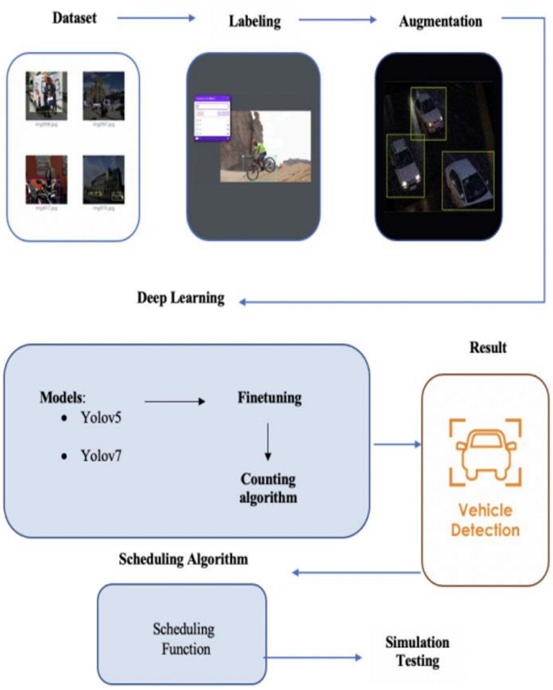

Figure 1. The proposed system flow

3.2 Queue length-based scheduling method

The queue length–based method is particularly effective for estimating green times in scenarios where the primary objective is to clear existing queues, rather than coordinating vehicle arrivals to pass through the intersection during a continuous green phase. The fundamental assumption of this method is that the required green time is proportional to the number of vehicles waiting in the queue. When a queue forms, each vehicle requires a certain average discharge time to cross the intersection once the signal turns green. Therefore, clearing a queue of length QL_i on road i requires a green time that accounts for both (a) Start-up lost time, which represents the delay caused by driver reaction time and vehicle acceleration at the beginning of the green phase, and (b) Per-vehicle crossing time. The following equation is used to calculate the green time for a lane on road i based on its Queue Length QL:

$$GST_i = StartupLostTime + (QL_i \times AvgTimePerVehicleClass) \quad (1)$$

This formulation ensures that longer queues are allocated proportionally longer green signal durations, allowing queued vehicles to be discharged efficiently. Following the same process, the green time is also calculated for the adjacent road (*i+1*). In the proposed system, queue lengths are capped at a predefined maximum to prevent excessive green durations and to maintain fairness among competing approaches. The queue length is then classified as either medium or long for road *i*, and the same classification is applied to the adjacent road (*i+1*). Subsequently, the conditions shown in Figure 2 are checked to determine the appropriate scheduling.

Input:
- GST_i, GST_{i+1}, green time signals for road i and its adjacent road $i+1$
- QL_i, QL_{i+1}, queue lengths for the roads

For i from 1 to 3:
  IF $QL_i = QL_{i+1}$
    $GST_i = GST_i/2$
  ELSE IF $QL_i < QL_{i+1}$:
    $GST_i = GST_i - (GST_{i+1} - GST_i)$ //reduce GST_i by the green time difference
  ELSE IF $QL_i > QL_{i+1}$:
    $GST_i = GST_i - (QL_i - QL_{i+1})$ //reduce GST_i by the queue length difference

Figure 2. Conditions used in the *setTime* function to determine the appropriate scheduling for the Queue Length-based method

Unlike traditional queue-based methods that consider each road independently, the proposed approach compares queue conditions between adjacent roads. As shown in Figure 2, this scheduling method determines the green signal duration by comparing the queue lengths of two adjacent roads at an intersection. Initially, the queue length for each road is estimated by counting the number of vehicles waiting at the stop line during the vehicle detection phase. If both road *i* and road (*i+1*) exhibit similar queue length, the algorithm updates the GTS_i to half of the computed green time. When road *i* has a shorter queue than road (*i+1*), the algorithm decreases GTS_i by the green time difference $GTS_{i+1} - GTS_i$. Conversely, if road *i* has a longer queue than road *i+1*, the algorithm reduces GTS_i of the current road by the disparity in queue lengths $QL_i - QL_{i+1}$, which is a smaller difference than the first case to avoid any coming congestions in road *i+1*.

3.3 Density ratio-based scheduling method

While queue length captures instantaneous congestion, it does not fully reflect the overall traffic demand across the entire intersection. To address this limitation, the Density Ratio-Based method is introduced, which utilizes a modified density equation adapted from Gandhi et al. [9], drawing from traffic density theory and proportional resource allocation principles. In multi-approach intersections, optimal signal timing requires coordinating green times among all approaches rather than serving the most congested lane in isolation. The algorithm considers multiple factors, including the number of lanes, the total number of vehicles of each class (e.g., cars, buses, trucks, and motorcycles), traffic density, start-up latency, average vehicle speeds, and minimum and maximum green-light duration limits. The total green signal time (GST) required for a road is computed by summing the weighted crossing times of all detected vehicles as follows:

$$GST = \frac{\sum_{vClass}(NoOfVehicles_{vClass} \times AvgTime_{vClass})}{(NoOfLanes+1)} \tag{2}$$

This formulation assigns greater weight to heavier or slower vehicles, ensuring realistic green-time estimation. In Eq. (2), GST denotes the green signal time, $NoOfVehicles_{vClass}$ represents the number of vehicles of each class detected at the signal, $AvgTime_{vClass}$ is the average time required for vehicles of that class to cross the intersection, and $NoOfLanes$ is the total number of lanes at the intersection.

Green times are assigned to one signal, while the corresponding red times are allocated to the adjacent signal. The system follows a predictable sequential signal transition, avoiding prioritization of the most congested direction to remain consistent with existing traffic patterns. Yellow signals and their transitions are also incorporated to ensure smooth and safe traffic flow.

The proposed system calculates GSTs by analyzing the total number of vehicles on all roads at an intersection. The system dynamically adjusts green times based on real-time traffic data to balance traffic flow in all directions. Accordingly, the total cycle time of the intersection is calculated as the sum of the GSTs allocated to each road [9].

$$CycleTime = \sum_{i=1}^{4} GST_i \tag{3}$$

To ensure fair allocation, the average green time per cycle is computed as shown in Eq. (4), which accounts for both the cycle duration and the total yellow signal time.

$$avGreen = (CycleTime / 4) + TotalYellowTime \tag{4}$$

Finally, the green signal duration for each road is proportionally assigned based on its traffic density relative to the entire intersection, as formulated in Eq. (5).

$$GTS_i = \frac{NoOfVehicles_i}{TotalVehicles} \times avGreen \tag{5}$$

By distributing green time according to density ratios, the system adapts to uneven traffic distributions and improves global intersection throughput rather than local queue clearance. The minimum and maximum green times are set to 10 and 60 seconds, respectively. This ensures the system remains responsive to traffic conditions while avoiding excessive delays. Table 1 illustrates a sample calculation of green light time for road 1 based on the density ratio. Roads 2 and 3, having higher vehicle densities, reduce green time for road 1 from 17 seconds to 11 seconds, as calculated by:

$$GTS_1 = (22/71) \times 34 = 11\,seconds \tag{6}$$

This dynamic approach improves traffic flow and reduces congestion by optimizing the green light duration for each road.

Table 1. Example of computing green signal time (GST) for the density ratio-based method

Road	No. of Vehicles	GTS (sec)	Density Ratio-Based GTS
1	22	17	11
2	27	19	13
3	15	11	7
4	7	6	4
Total	71	53	35

The queue length–based and density ratio–based methods address different aspects of traffic congestion. The queue-based approach focuses on short-term queue clearance, making it effective during sudden congestion buildup. In contrast, the density ratio–based method provides a global and proportional view of intersection demand, leading to smoother and more balanced traffic flow over time.

4. EXPERIMENTAL SETUP

This section outlines the end-to-end experimental workflow followed in this study. We first introduced the dataset preparation for the vehicle detection model in subsection 4.1, then we describe the adopted assessment methods for the vehicle detection module and the simulation-based performance in subsection 4.2.

Figure 3. Model predictions on validation images

4.1 Data acquisition and preparation

The dataset used in this study is a published vehicle dataset [28], representing five main types of vehicles, namely, car, motorcycle, truck, bus, and bicycle. The original dataset contained 1,251 images with a resolution of 640 × 640 pixels. These images were divided into training, validation, and testing sets in a 70:20:10 ratio, resulting in 886 images for training, 243 for validation, and 122 for testing. To address biases in the dataset and the limited number of images for the "truck" and "bicycle" classes, 100 additional annotated images were added to these categories. These annotations were created using Roboflow1 and exported in the YOLO format. Additionally, various image augmentation techniques were applied to enhance the dataset, including flipping, cropping, adjusting contrast, and adding noise. These efforts resulted in an updated dataset comprising a total of 9,145 images: 6,400 for training, 1,800 for validation, and 912 for testing. It is worth noting that the dataset includes variability in vehicle types, lighting conditions, and camera angles. However, the dataset does not fully capture rare events such as roadworks and extreme weather. Figure 3 shows an example of a YOLO model prediction on selected validation images.

4.2 Evaluation methods

The proposed smart traffic light network was evaluated to test the effectiveness of the YOLO models and the signal scheduling algorithm. First, the detection model is evaluated using several metrics to get useful information about the system's performance. The confusion matrix was utilized to examine true positive, false positive, false negative, and true negative predictions for each type of detected vehicle, offering a detailed summary of classification performance together with a visual representation of the model's accuracy and misclassifications. These metrics are accuracy, precision, recall, F1-score, and mean average precision (mAP). Accuracy was measured to determine the proportion of right predictions out of all predictions made, which is essential for assessing the detection system's overall efficiency. Precision, which assesses the accuracy of the detection, guarantees that the model does not generate many false positives. Recall, also called sensitivity, measures how well the model can correctly recognize all relevant instances. While the F1- score is used to balance precision and recall, the mean average precision (mAP) is used to examine the precision of the bounding boxes generated by the YOLO models across different types of vehicles.

The proposed scheduling algorithm was evaluated using a Pygame-based simulation environment modeling a four-way signalized intersection (as shown in Figure 4). The simulator, adapted from the implementation described by Gandhi et al. [9], was used to assess system performance under controlled and reproducible traffic conditions. Each simulation run was executed for a fixed duration of five minutes, during which vehicles of different types—including cars, buses, trucks, motorcycles, and bicycles—entered the intersection from all approaches according to predefined traffic distributions. A timer also shows the elapsed time since the simulation began. The scheduling process starts during the last 5 seconds of the yellow light, detecting and counting all vehicles in the lane. This continues until it measures the adaptive green light duration. To improve efficiency and optimize the green light duration based on the simulation, the *setTime* method used to establish the green duration for the intersection has been modified and enhanced by the scheduling algorithm.

The primary evaluation metric was intersection throughput, measured as the total number of vehicles that successfully passed through the intersection within the simulation time window. Performance was evaluated under multiple traffic scenarios, including balanced, moderately skewed, and sharply skewed traffic distributions, and compared against baseline approaches consisting of static signal timing and the adaptive method reported by Gandhi et al. [9]. All experiments were conducted using identical simulation parameters and traffic arrival patterns to ensure fair comparison.

Figure 4. Pygame-based traffic simulation environment for a 4-way intersection

4.3 Implementation settings

For the vehicle detection experiments, YOLOv5m (https://github.com/ultralytics/yolov5) (41 million parameters) and YOLOv7 (https://github.com/ultralytics/yolov5) models were selected and initialized using pretrained weights, which were then fine-tuned on the described dataset. All models were trained using an input image resolution of 640 × 640 pixels. Training was conducted for 20 epochs, as preliminary experiments indicated that performance converged within this range. The batch size is set to 4 with the SGD optimizer and 0.01 learning rate. All experiments were conducted on Google Collaboratory, utilizing its cloud-based GPU resources. For traffic signal control, the maximum queue length per lane was capped at ten vehicles to prevent excessive green signal durations and maintain stable signal cycles. The minimum (10) and maximum (60) green signal durations were constrained to ensure realistic signal operation and to avoid rapid phase switching.

5. EXPERIMENTAL RESULTS

The evaluation of this study was conducted in two phases. First, the performance of the detection models: YOLOv5 and YOLOv7, was evaluated and compared with baselines of recent YOLO versions trained on the same vehicle dataset. Second, the scheduling algorithm was evaluated and compared to previous research works. These evaluations aimed to assess the system's ability to effectively manage traffic congestion using dynamic scheduling and detection techniques.

5.1 Vehicle detection model evaluation

Figures 5 and 6 illustrate the training and validation performance of the YOLOv5 and YOLOv7 models, respectively, over the training epochs. Both models exhibit consistent convergence behavior, as evidenced by the monotonic decrease in box loss and classification loss during training and validation phases. This trend indicates stable learning and effective optimization for vehicle detection tasks. The precision and recall metrics show steady improvement throughout training, reaching high values by the final epochs. Additionally, the mAP@0.5 metric approaches near-optimal performance, while mAP@0.5:0.95 increases more gradually, reflecting improved localization accuracy under stricter evaluation criteria.

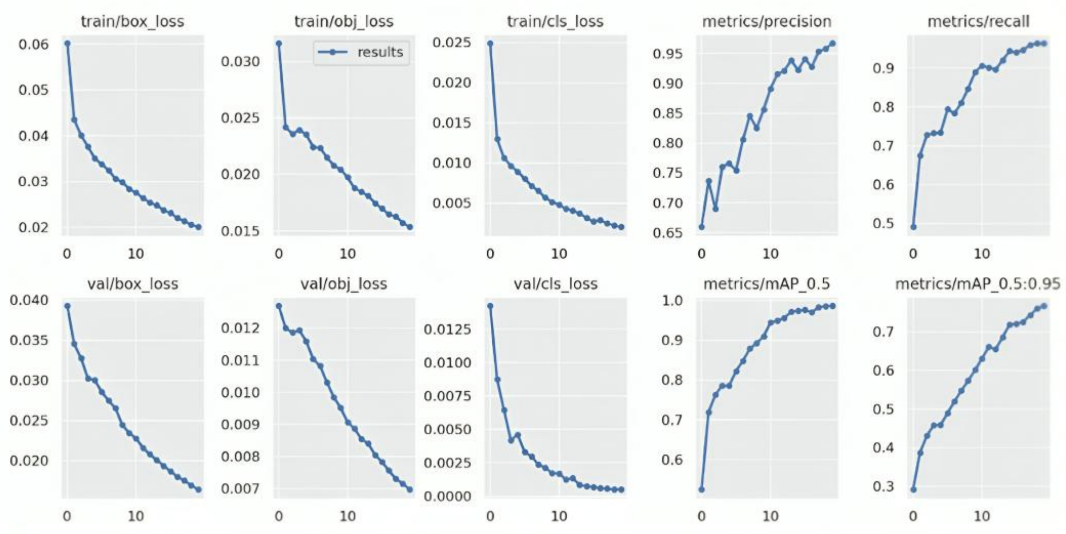

Figure 5. Training and validation losses of the YOLOv5

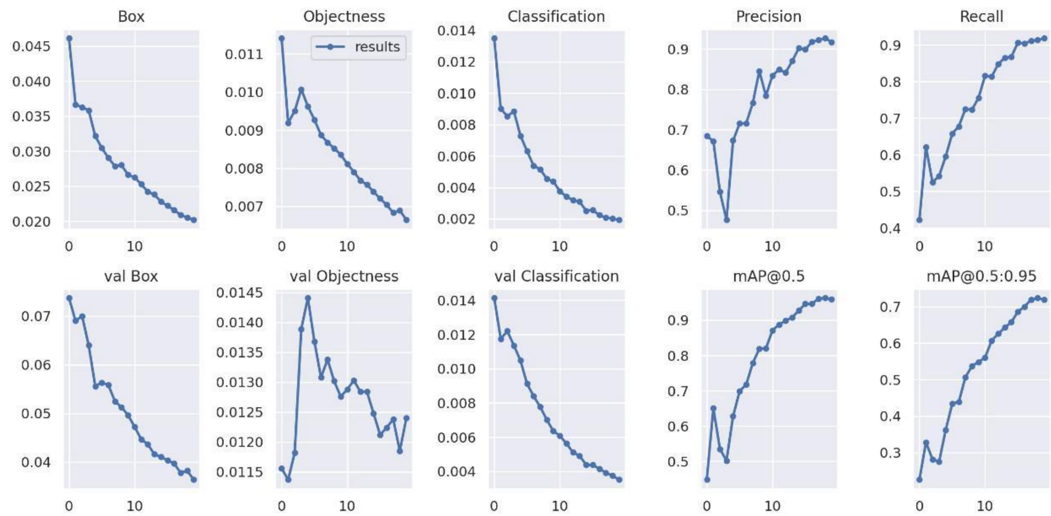

Figure 6. Training and validation losses of the YOLOv7

In Table 2, precision confidence and recall confidence curves indicate high precision (100%) and recall (99%), with corresponding confidence levels of 0.971 and 0.0 across all classes. The model achieved mAP of 98.6% at a 0.5 threshold, reflecting strong performance in bounding box detection. Additionally, the F1 score is 96%, demonstrating a balanced trade-off between precision and recall. For YOLOv7, Table 2 presents precision (100%) and recall (99%) values that are similarly high, with confidence levels of 0.992 and 0.0. YOLOv7 achieved mAP of 95.9% at a 0.5 threshold, indicating robust detection performance, while the F1 score stands at 92%, reflecting an appropriate trade-off between precision and recall. This comparative analysis of YOLOv5 and YOLOv7 shows that both models achieved high precision and recall. However, YOLOv5 outperformed YOLOv7 in terms of mAP (98.6% vs. 95.9%) and F1 score (96% vs. 92%). YOLOv5's superior performance is attributed to its optimized design for standard GPUs, while YOLOv7, which performs better on high-speed GPUs, has slower training speeds on custom datasets due to its higher computational demands.

For YOLOv5, the normalized confusion matrix in Figure 7 shows true positive values along the diagonal, with false positives and false negatives off the diagonal. YOLOv5 shows strong performance in correctly classifying all vehicle classes with a high accuracy of 0.99 for Motorcycle and Bus. For trucks, the model achieved correct classification scores of 0.96. The confusion matrix confirms high classification accuracy with minimal cross-class confusion.

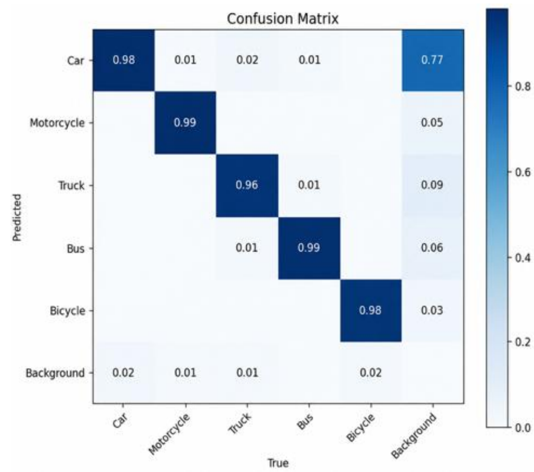

Figure 7. YOLOv5 normalized confusion matrix

Table 2. Comparative analysis of YOLOv5 and YOLOv7

Model	Precision	Recall	F1	mAP@0.5
YOLOv5	100	99	96	98.6
YOLOv7	100	99	92	95.9

Figure 8 shows recall-precision and confidence-F1 curves for YOLOv5m and YOLOv7. The F1–confidence curves depict the harmonic mean of the precision and recall at various confidence thresholds for different vehicle classes. For YOLOv5m, the F1 score for all classes peaked at 0.96 at a confidence level of 0.484, indicating a balanced trade-off between recall and precision at this threshold. In contrast, YOLOv7 achieved a slightly lower peak F1 score of 0.92 at the confidence level 0.391, reflecting degradation in the performance compared with YOLOv5m. Similar improvements are shown for YOLOv5 from the Recall-Precision curves for all the classes.

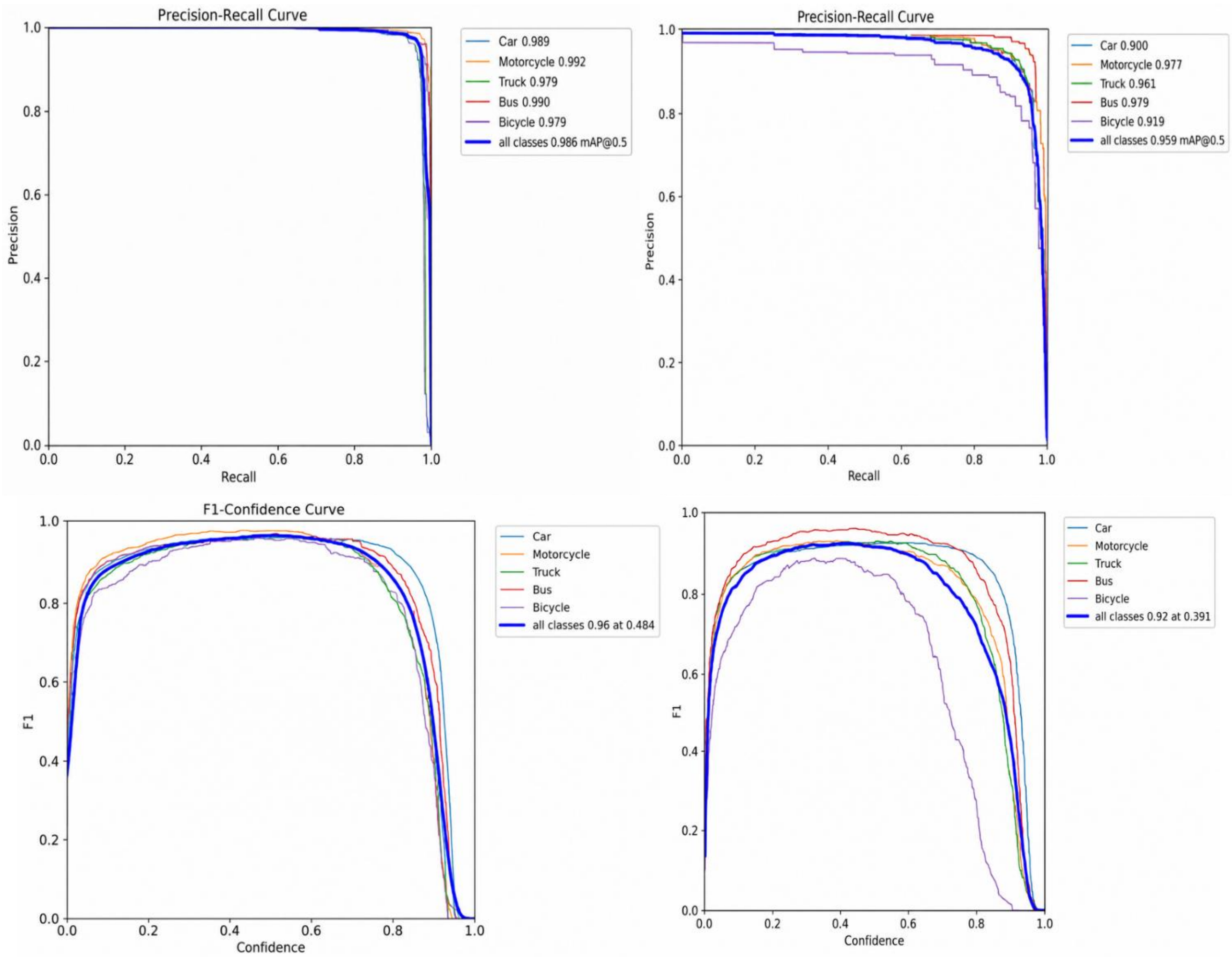

Figure 8. Recall-precision and confidence-F1 curves for YOLOv5m (left column) and YOLOv7 (right column)

Table 3. Comparison of per-class recall and macro-average recall for our best model (YOLOv5) and other fine-tuned models from the literature on the same vehicle dataset

Model	Car	Bus	Truck	Bicycle	Motorcycle	Macro Av.
YOLOv8 [28]	0.76	0.89	0.33	0.62	0.63	0.65
YOLOv10 [28]	0.75	0.89	0.33	0.62	0.74	0.67
YOLOv5 (our)	0.98	0.99	0.96	0.98	0.99	0.98

Comparisons Against Baselines. We compare our best obtained results from YOLOv5 with recent work that adopts advanced YOLO versions (YOLOv8 and YOLOv10) for the same vehicle dataset [29]. Table 3 shows the comparison of the diagonal values (True Positives) from normalized confusion matrices for all vehicle classes (since it has been reported in the previous research). Per-class recall extracted from the normalized confusion matrices shows that YOLOv5 consistently outperforms YOLOv10 across all object categories, with a macro-averaged recall of 0.978 vs 0.666. The largest degradation is observed for the Truck class (−0.62). Comparing the off-diagonal entries of the confusion matrices shows that YOLOv10 exhibits systematic confusion between visually similar classes (truck–car), whereas YOLOv5 maintains class-separable predictions.

5.2 Scheduling algorithm evaluation

The Scheduling Algorithm was evaluated through a Pygame-based simulation of a 4-way intersection [9]. The simulation visually represents traffic flow, with each signal displaying a timer for green, yellow, and red transitions, and tracking intersection throughput. Vehicles include cars, bikes, buses, trucks, and others, with some vehicles turning at intersections. The *setTime* algorithm dynamically adjusts green light durations based on vehicle counts during the yellow light phase, enhancing efficiency. We conducted a simulation of 15 different experiments with various vehicle distributions across the four roads of the intersection. To demonstrate the advantages and applicability of the proposed traffic signal control system, the experimental results were analyzed in comparison with traditional fixed-time control and single-road density–based adaptive methods.

Queue Length Algorithm Evaluation. The *setTime* function was modified to dynamically allocate green light durations based on the queue lengths of adjacent roads (road i and road i + 1). The algorithm adjusts light timings to prioritize roads with longer queues, improving vehicle throughput. The performance is shown in Figure 9, where the X-axis corresponds to the simulation experiment, each representing a traffic distribution scenario described in Table 2, while the Y-axis represents the number of vehicles that successfully crossed the intersection. Over 15 experiments, each lasting 5 minutes with varying vehicle distributions, the queue-based method significantly improved vehicle flow compared to static timing and the single road density method proposed by Gandhi et al. [9]. The approach resulted in 4-5 full cycles within the 5-minute period, optimizing traffic flow and minimizing delays.

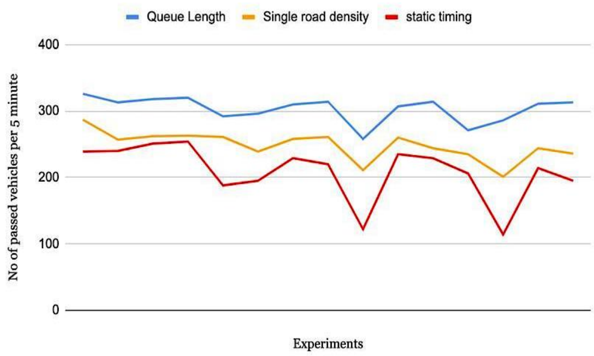

Figure 9. Queue length optimization

Density Ratio Algorithm Evaluation. In the density ratio algorithm, the *setTime* function adjusts the green light times based on the density ratio of road i relative to the overall density of the intersection. This method outperformed both the prior system and the static timing system under various traffic conditions. The algorithm reduces green light times for roads with smaller density ratios and allocates more time to roads with larger density ratios. This dynamic adjustment optimizes traffic flow as illustrated in Figure 10, increasing intersection throughput.

Queue Length vs Density Ratio. Both the queue length and density ratio algorithms showed significant improvements over existing models. However, the density ratio algorithm slightly outperformed the queue length algorithm as it considers the entire intersection, leading to better overall vehicle flow (as shown in Figure 11). The density ratio approach is more dynamic, reacting to short-term demand changes and minimizing excessive delays.

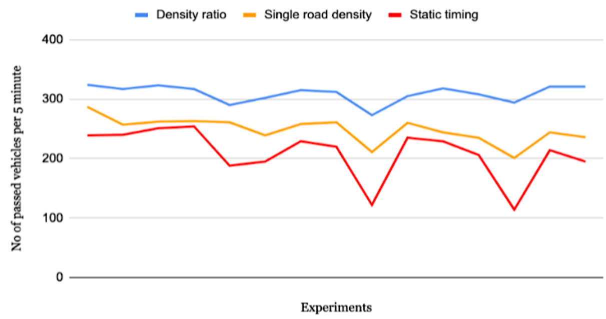

Figure 10. Density ratio algorithm evaluation

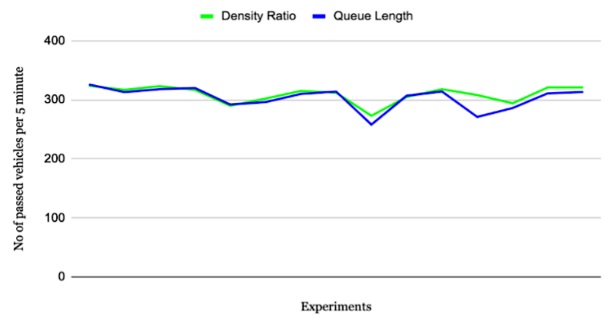

Figure 11. Density ratio vs. queue length

Table 4. Experiments with Pygame simulations. #C denotes the number of full cycles in the 5 minutes, L1 to L3 are the number of vehicles in the corresponding lane

Ex	#C	Distributions	L1	L2	L3	L4
1	4	[300,600,300,1000]	95	86	65	78
2	4	[500,700,900,1000]	149	71	64	33
3	4	[250,500,750,1000]	90	83	76	74
4	4	[300,500,800,1000]	104	55	103	55
5	3	[700,800,900,1000]	199	21	41	29
6	3	[500,900,950,1000]	138	123	25	16
7	4	[300,600,900,1000]	92	94	87	42
8	3	[200,700,750,1000]	95	145	14	94
9	3	[940,960,980,1000]	254	4	10	5
10	3	[400,500,900,1000]	104	26	135	40
11	4	[200,400,600,1000]	48	67	61	142
12	3	[250,500,950,1000]	73	64	151	20
13	5	[850,900,950,1000]	252	10	15	17
14	5	[350,500,850,1000]	113	55	104	49
15	4	[360,700,850,1000]	109	97	62	53

Table 5. Improvement percentages of the proposed queue-length and density ratio against baseline for simulations of equal, moderate, and sharply skewed distributions

Density Ratio and Static Timing	Density Ratio and Single Density	Queue Length and Static Timing	Queue Length and Single Density
	Equal Distribution		
33%	22%	31%	20%
	Moderately Skewed		
50%	24%	47%	22%
	Sharply Skewed		
140%	37%	130%	32%

Comparisons Against Baselines. As shown in Table 4, the distribution of traffic varied across experiments, ranging from equal distribution (experiments 1–4, 10–12) to moderately skewed (experiments 5–8, 14–15) or sharply skewed (experiments 9 and 13) distributions. Table 5 illustrates that the improvements were more significant in cases with skewed distributions, as the dynamic algorithms could better handle such scenarios, resulting in a higher percentage increase in performance. In particular, under sharply imbalanced traffic conditions, the proposed method increases throughput by up to 140% relative to a static signal control system and by 37% compared to the adaptive method reported by Gandhi et al. [9]. These gains are achieved by considering congestion across the entire intersection, enabling more efficient green time allocation and reduced red signal durations on heavily congested approaches.

6. DISCUSSION

6.1 End-user admin system

It is important to evaluate the real-world applicability of our proposed traffic signaling detection from the stakeholder perspectives. While the primary focus of this work is on algorithmic validation through the fine-tuned vehicle detection model and the simulation-based evaluation of the proposed green-time calculation strategy, we also designed an administrative monitoring system (called JAZZ (JAZ is an Arabic name that means 'passing across' or 'making a transit' and is also used to signify 'being allowed', 'being permitted,' and 'saying Jaz')) to support stakeholder interaction and post-deployment evaluation. The developed admin system enables the centralized management, storage, and visualization of traffic data generated by the proposed system. Specifically, it records operational metrics such as the number of vehicles detected per lane, green and red signal durations for each cycle, vehicle flow rates, and intersection throughput over time. These data are stored in a dedicated database and made accessible through an administrative interface for traffic engineers and decision-makers. Figures 12 and 13 illustrate screenshots of the proposed system and its main functionalities.

This monitoring platform serves as an indirect yet practical mechanism for collecting stakeholder feedback by allowing transportation authorities and traffic operators to analyze system behavior, evaluate performance trends, and identify inefficiencies under real traffic conditions. By examining historical and real-time traffic indicators, stakeholders can provide informed feedback on signal timing effectiveness, congestion reduction, and operational reliability.

In future deployments, this administrative system can be extended to incorporate explicit stakeholder feedback mechanisms, such as performance evaluation reports, operator annotations, and user surveys, further strengthening the practical applicability of the proposed solution.

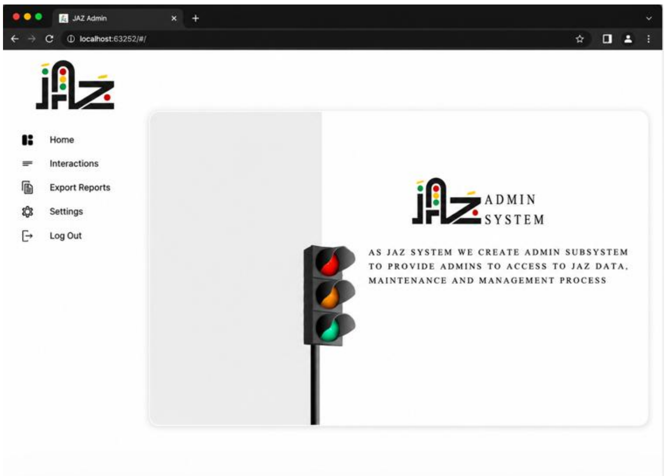

Figure 12. The main interface of the JAZZ admin system

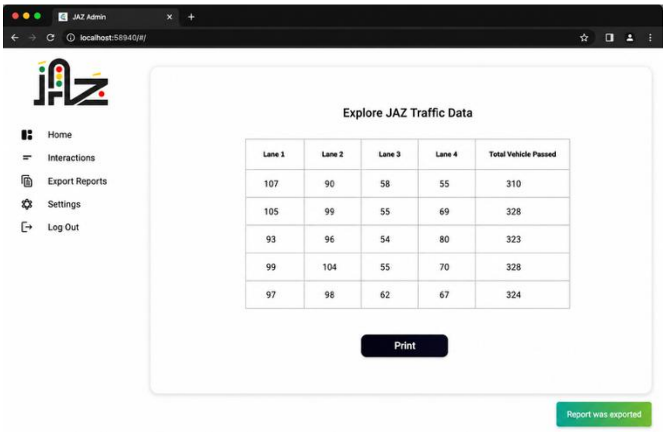

Figure 13. Exploring the JAZZ traffic data interface

6.2 Generalizability of the proposed model

While the vehicle dataset used in this study includes multiple vehicle classes and captures variability in lighting conditions and viewpoints, it does not encompass all possible real-world traffic scenarios, such as extreme weather conditions and roadwork. This limitation is acknowledged as a constraint of the current experimental setup for the vehicle detection model. However, the generalizability of the proposed system is primarily supported by the modular separation between the vehicle detection component and the traffic signal scheduling algorithms. The proposed queue length–based and density ratio–based signal timing methods rely on aggregate traffic indicators (e.g., vehicle counts, queue lengths, and density ratios) rather than dataset-specific features. As a result, the scheduling algorithms remain applicable regardless of the specific detection model or dataset employed, provided that reliable vehicle counts can be obtained.

To further enhance generalizability, future work will focus on evaluating the system using multi-city datasets that capture a broader range of traffic conditions, including adverse weather, peak-hour congestion, and atypical traffic events. Additionally, real-world pilot deployments will be considered to validate system performance under operational conditions and to assess robustness across diverse urban environments.

6.3 Real-world applicability: Case studies

To demonstrate the practical applicability of the proposed traffic signal control system, this section presents representative case studies that illustrate how the method can be deployed and operated in real-world urban environments. Although the current evaluation is conducted in a simulation environment, the system design is motivated by realistic traffic management requirements and existing infrastructure constraints.

Consider a signalized four-way intersection in a metropolitan area equipped with roadside surveillance cameras. During morning and evening peak hours, traffic demand becomes uneven across approaches due to commuter flow patterns. In this scenario, the proposed vehicle detection module processes live video streams to estimate queue lengths and traffic densities in real time. The queue length–based method prioritizes heavily congested approaches to prevent queue spillback, while the density ratio–based method ensures balanced green time allocation across the entire intersection. Traffic engineers can monitor system performance through the administrative interface and adjust operational parameters as needed.

In many urban environments, traffic consists of heterogeneous vehicle types, including cars, buses, trucks, motorcycles, and bicycles. The proposed system explicitly accounts for vehicle classes when computing green signal durations, allowing heavier or slower vehicles to receive adequate crossing time. This capability is particularly beneficial near transit hubs or commercial districts, where large vehicles frequently contribute to congestion.

6.4 Environmental impact of the proposed model

Traffic signal timing plays a critical role in shaping transportation system performance and has its direct implications for mobility, energy consumption, and environmental sustainability. Inefficient signal control contributes to increased vehicle idling, frequent stop-and-go driving, and prolonged travel times, all of which lead to higher fuel consumption and elevated emissions of greenhouse gases and air pollutants.

The proposed smart adaptive traffic light system addresses these challenges by dynamically adjusting signal timings in response to real-time traffic conditions. By reducing unnecessary red signal durations, the system minimizes vehicle idle time and improves traffic flow efficiency at intersections. These improvements directly contribute to lower fuel consumption and reduced emissions, including carbon dioxide (CO_2), thereby supporting cleaner urban air quality.

From a sustainability perspective, the proposed approach supports smart city efforts to improve energy efficiency and reduce the environmental impact of urban traffic systems. Unlike hardware-intensive solutions, the system relies on existing traffic cameras and software-based processing, which limits additional material requirements and reduces energy consumption and maintenance needs. As a result, the approach is suitable for cost-effective and scalable deployment.

Furthermore, by integrating artificial intelligence and computer vision into traffic management, the proposed system supports long-term urban sustainability goals such as reduced traffic congestion, improved travel reliability, and enhanced quality of life. These objectives are consistent with national and regional smart city visions, where intelligent, data-driven transportation systems are essential for achieving sustainable and environmentally responsible urban growth.

7. CONCLUSION

This study utilizes the power of AI techniques to introduce a smart adaptive traffic signal system as a novel way to address the issue of traffic congestion. The primary function of the suggested system is to intelligently adjust traffic light signal durations in response to other lanes in the intersection and road conditions, such as the number and type of vehicles detected at any given time in the entire intersection and the state of the road in terms of congestion.

The proposed system leverages computer vision and machine learning techniques to estimate traffic density using real-time images captured from traffic junction cameras. Signal phase durations, including green time, red time, and transition intervals, are computed using a mathematical traffic control model. We report high performance of vehicle detection models based on fine-tuning small versions of YOLO-based models. The simulation results indicate improved intersection throughput under both balanced and unbalanced traffic distributions. These gains are achieved by considering congestion across the entire intersection, enabling more efficient green time allocation and reduced red signal durations on heavily congested approaches.

REFERENCES

[1] Downs, A. (2000). Stuck in Traffic: Coping with Peak-Hour Traffic Congestion. Bloomsbury Publishing USA.

[2] Aftabuzzaman, M. (2007). Measuring traffic congestion- a critical review. In 30th Australasian Transport Research Forum. ETM GROUP London UK.

[3] Downs, A. (2005). Still Stuck in Traffic: Coping with Peak-Hour Traffic Congestion. Rowman & Littlefield. https://doi.org/10.1080/00420980500332155

[4] Wandhare, M.U.R., Yawale, M.A., Rathod, M.A., Meshram, M.A. (2020). Case study of traffic congestion. International Research Journal of Engineering and Technology, 7(2): 2913.

[5] Retallack, A.E., Ostendorf, B. (2019). Current understanding of the effects of congestion on traffic accidents. International Journal of Environmental Research and Public Health, 16(18): 3400. https://doi.org/10.3390/ijerph16183400

[6] Mapon. (2023). 20 amazing facts about traffic and traffic jams. https://www.mapon.com/en/blog/20-amazing-facts-about-traffic-and-traffic-jams, accessed on Jan. 2,

2026.
[7] Riyadh Traffic. (2025). Riyadh traffic report | TomTom Traffic Index. https://www.tomtom.com/traffic-index/riyadh-traffic.
[8] Mohamed, N.E., Radwan, I.I. (2022). Traffic light control design approaches: A systematic literature review. International Journal of Electrical & Computer Engineering, 12(5): 5355. https://doi.org/10.11591/ijece.v12i5.pp5355-5363
[9] Gandhi, M.M., Solanki, D.S., Daptardar, R.S., Baloorkar, N.S. (2020). Smart control of traffic light using artificial intelligence. In 2020 5th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE), Jaipur, India, pp. 1-6. https://doi.org/10.1109/ICRAIE51050.2020.9358334
[10] Mahato, S. (2025). Smart traffic signals: Comparing MARL and fixed-time strategies. arXiv preprint arXiv:2505.14544. https://doi.org/10.48550/arXiv.2505.14544
[11] Yalçınlı, F., Akdemir, B., Durdu, A., Ali, E., Akkaş, K.Ö. (2025). Signaling improvement model and efficiency analysis of the model. Traitement du Signal, 42(2): 689. https://doi.org/10.18280/ts.420208
[12] Firdous, A., Niranjan, V. (2020, June). Smart density based traffic light system. In 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, pp. 497-500. https://doi.org/10.1109/ICRITO48877.2020.9197940
[13] Alaidi, A., Aljazaery, I., Alrikabi, H., Mahmood, I., Abed, F. (2020). Design and implementation of a smart traffic light management system controlled wirelessly by arduino. International Association of Online Engineering 14(7): 32. https://www.learntechlib.org/p/216461/.
[14] Atta, A., Abbas, S., Khan, M.A., Ahmed, G., Farooq, U. (2020). An adaptive approach: Smart traffic congestion control system. Journal of King Saud University-Computer and Information Sciences, 32(9): 1012-1019. https://doi.org/10.1016/j.jksuci.2018.10.011
[15] Abishek, C., Kumar, M., Kumar, P. (2009). City traffic congestion control in Indian scenario using wireless sensors network. In 2009 Fifth International Conference on Wireless Communication and Sensor Networks (WCSN), Allahabad, India, pp. 1-6. https://doi.org/10.1109/WCSN.2009.5434809
[16] Salama, A.S., Saleh, B.K., Eassa, M.M. (2010). Intelligent cross road traffic management system (ICRTMS). In 2010 2nd International Conference on Computer Technology and Development, Cairo, Egypt, pp. 27-31. https://doi.org/10.1109/ICCTD.2010.5646059
[17] Zhao, H., Zheng, X., Liu, W. (2009). Intelligent traffic control system based on DSP and NIOS II. In 2009 International Asia Conference on Informatics in Control, Automation and Robotics, Bangkok, Thailand, pp. 90-94. https://doi.org/10.1109/CAR.2009.59
[18] Narula, M.S., Ayush, A., Kumar, M. (2024). Advanced traffic light controller using FPGA and ARDUINO. Journal of ISMAC, 6(2): 149-164. https://doi.org/10.36548/jismac.2024.2.006
[19] Albatish, I.M., Abu-Naser, S.S. (2019). Modeling and controlling smart traffic light system using a rule based system. In 2019 International Conference on Promising Electronic Technologies (ICPET), Gaza, Palestine, pp. 55-60. https://doi.org/10.1109/ICPET.2019.00018
[20] Yousef, K.M.A., Shatnawi, A., Latayfeh, M. (2019). Intelligent traffic light scheduling technique using calendar-based history information. Future Generation Computer Systems, 91: 124-135. https://doi.org/10.1016/j.future.2018.08.037
[21] Razavi, M., Hamidkhani, M., Sadeghi, R. (2019). Smart traffic light scheduling in smart city using image and video processing. In 2019 3rd International Conference on Internet of Things and Applications (IoT), Isfahan, Iran, pp. 1-4. https://doi.org/10.1109/IICITA.2019.8808836
[22] Khekare, G.S., Sakhare, A.V. (2013). A smart city framework for intelligent traffic system using VANET. In 2013 International Mutli-Conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s), Kottayam, India, pp. 302-305. https://doi.org/10.1109/iMac4s.2013.6526427
[23] Badura, S., Lieskovsky, A. (2010). Intelligent traffic system: Cooperation of MANET and image processing. In 2010 First International Conference on Integrated Intelligent Computing, Bangalore, India. pp. 119-123.
[24] Kanungo, A., Sharma, A., Singla, C. (2014). Smart traffic lights switching and traffic density calculation using video processing. In 2014 Recent Advances in Engineering and Computational Sciences (RAECS), Chandigarh, India, pp. 1-6. https://doi.org/10.1109/RAECS.2014.6799542
[25] Bouktif, S., Cheniki, A., Ouni, A. (2021). Traffic signal control using hybrid action space deep reinforcement learning. Sensors, 21(7): 2302. https://doi.org/10.3390/s21072302
[26] Li, M., Pan, X., Liu, C., Li, Z. (2025). Federated deep reinforcement learning-based urban traffic signal optimal control. Scientific Reports, 15(1): 11724. https://doi.org/10.1038/s41598-025-91966-1
[27] Saadi, A., Abghour, N., Chiba, Z., Moussaid, K., Ali, S. (2025). A survey of reinforcement and deep reinforcement learning for coordination in intelligent traffic light control. Journal of Big Data, 12(1): 84. https://doi.org/10.1186/s40537-025-01104-x
[28] Boneh, M. (2025). Vehicle detection, 2024. https://github.com/MaryamBoneh/Vehicle-Detection.
[29] Sundaresan Geetha, A., Alif, M.A.R., Hussain, M., Allen, P. (2024). Comparative analysis of YOLOv8 and YOLOv10 in vehicle detection: Performance metrics and model efficacy. Vehicles, 6(3): 1364-1382. https://doi.org/10.3390/vehicles6030065