The International Society of Data Scientists
Journal of Data Science and Artificial Intelligence
Vol. 5, No. 1. Online ISSN: 2831-4794

An Adaptive ROI-Based Framework for Traffic Congestion Detection in Mixed Traffic Environments Using Deep Learning

Ngo Duong Ha[1], Truong My Hoa[1], Luong Tan Tai[1], Cao Thien Chi[1], Tran Nhu Y[1]*
[1]Faculty of Information Technology, Ho Chi Minh City University of Industry and Trade, 700000, Vietnam
*Corresponding author. Email: ytn@huit.edu.vn

ABSTRACT

Traffic congestion is a critical issue in urban areas, leading to increased travel time, fuel consumption, and environmental pollution. This paper proposes a real-time traffic congestion detection system based on deep learning and multi-object tracking. The system utilizes the YOLOv8 model for vehicle detection and ByteTrack for tracking vehicles across video frames. A region of interest (ROI) is defined to focus on relevant traffic areas, and congestion is determined using two key metrics: vehicle density and Motorcycle Equivalent Unit (MEU) - based traffic representation. Experimental results demonstrate that the proposed system achieves competitive detection accuracy, with a mean Average Precision (mAP@0.5) of 0.911 for vehicle detection. The system is able to distinguish between congested and non-congested traffic conditions in real-world scenarios. These results indicate that the proposed approach is suitable for intelligent traffic monitoring and smart city applications.

Keywords: Congestion Detection; YOLOv8; Multi-Object Tracking; ROI; Motorcycle Equivalent Unit.

1. INTRODUCTION

Traffic congestion has become a critical challenge in modern urban environments, particularly in rapidly developing cities where the number of vehicles increases significantly over time. In large metropolitan areas such as Ho Chi Minh City, traffic congestion frequently occurs at major intersections during peak hours, leading to severe consequences including increased travel time, fuel consumption, air pollution, and reduced quality of life [1], [2]. The complexity of urban traffic systems, especially in developing countries, is further intensified by the heterogeneous nature of traffic, where motorcycles dominate the traffic flow alongside cars, buses, and trucks [3].

Traditional traffic monitoring systems rely on manual observation or simple sensor-based approaches, which are often limited in scalability, accuracy, and real-time responsiveness. Recently, advances in artificial intelligence (AI) and computer vision have enabled the development of intelligent transportation systems (ITS) capable of automatically analyzing traffic conditions from video data [4]. Object detection models such as YOLO [5] and Faster R-CNN [6] have been widely used to detect vehicles, while tracking algorithms such as SORT [7] and ByteTrack [8] allow for monitoring vehicle movement over time. These technologies provide a promising solution for real-time traffic analysis and congestion detection.

Despite these advancements, existing studies still exhibit several limitations. First, many approaches focus solely on image-level classification of congestion without explicitly detecting and tracking individual vehicles, which limits the ability to derive meaningful traffic parameters such as vehicle speed and density [9]. Second, some methods rely on optical flow techniques, which are sensitive to noise, camera vibration, and lighting variations [10]. Finally, most existing systems are designed for homogeneous traffic conditions and are not well suited for mixed traffic environments, particularly in countries like Vietnam where motorcycles constitute a large proportion of the traffic flow [3].

To address these challenges, this study proposes a real-time traffic congestion detection framework based on deep learning and multi-object tracking, specifically designed for mixed traffic environments. The proposed system integrates YOLOv8 [11] for vehicle detection and ByteTrack [8] for multi-object tracking, enabling accurate identification and continuous tracking of multiple vehicles in video streams. In addition, a Region of Interest (ROI)-based analysis is introduced to focus on specific areas of an intersection, allowing localized traffic assessment. To better represent heterogeneous traffic conditions, different types of vehicles are converted into a unified metric using the Motorcycle Equivalent Unit (MEU), which reflects their relative impact on traffic flow [3].

Based on the tracked vehicle data, the system computes key traffic indicators including vehicle density, average speed, and temporal stability. A congestion decision mechanism is then formulated using empirically determined thresholds derived from real-world observations. This approach enables the system to distinguish between normal traffic conditions and congestion states more effectively, particularly in dense and complex traffic scenarios.

The main contributions of this paper include the development of a real-time traffic congestion detection framework that combines YOLOv8 and ByteTrack for accurate vehicle detection and tracking, the introduction of an ROI-based analysis method for localized and flexible traffic monitoring at urban intersections, the proposal of a congestion evaluation model that integrates vehicle density, speed, and duration using MEU to handle mixed traffic environments, and the experimental validation on real-world traffic data demonstrating the effectiveness and applicability of the proposed system.

The remainder of this paper is organized as follows. Section 2 reviews related work in traffic congestion detection and computer vision-based approaches. Section 3 describes the proposed methodology in detail. Section 4 presents experimental results and performance evaluation. Finally, Section 5 concludes the paper and discusses future research directions.

2. RELATED WORK

Recent studies on traffic congestion detection from video data can generally be categorized into three main approaches: (i) image-level classification methods, (ii) motion- and speed-based analysis, and (iii) detection and tracking-based frameworks. Each category offers distinct advantages but also presents certain limitations, particularly when applied to complex and heterogeneous traffic environments.

In the first category, several studies formulate congestion detection as a holistic image classification problem using deep convolutional neural networks. For instance, the work “MobileNetV2 with Spatial Attention Module for Traffic Congestion Recognition” (2024) introduces a lightweight CNN architecture with spatial attention to enhance feature representation, achieving efficient performance suitable for resource-constrained deployment [12]. Similarly, Jiang and Feng (2025) propose a small parallel residual CNN to learn congestion-related features directly from images, demonstrating the effectiveness of specialized architectures for end-to-end classification [13]. To improve robustness, “Traffic Congestion Recognition Based on Convolutional Neural Networks in Different Scenarios” (2025) focuses on generalization across varying lighting and weather conditions [14]. Another study, “Traffic Congestion Detection from Camera Images Using Deep Learning Approach” (2024), employs traffic density classification as an intermediate representation for congestion detection [15]. While these approaches are computationally efficient and easy to train, they primarily rely on global image features and lack interpretability in terms of explicit traffic indicators such as vehicle count, speed, or temporal duration, limiting their applicability in real-world traffic management scenarios.

The second category emphasizes motion-based analysis by leveraging vehicle speed, optical flow, or temporal dynamics to characterize traffic conditions. FARSEC (2023) presents a reproducible framework for real-time vehicle speed estimation from traffic cameras, incorporating robustness against camera vibration and diverse video conditions [16]. Although not directly designed for congestion detection, it highlights the importance of speed as a key indicator of traffic states. Similarly, Ashraf et al. (2023) propose HVD-Net, which integrates detection, SORT-based tracking, and speed estimation, achieving approximately 87.24% accuracy in speed prediction [17]. Kamil et al. (2023) further demonstrate the effectiveness of combining YOLOv5 and DeepSORT for vehicle tracking and speed estimation, reporting RMSE and MAE values of 3.926 and 3.155, respectively [18]. Additionally, Azimjonov et al. (2023) develop a vision-based system capable of extracting traffic flow parameters such as vehicle count, movement direction, and speed, while analyzing the influence of camera placement [19]. Although these methods provide more dynamic and informative representations of traffic conditions, they often focus primarily on speed estimation or motion analysis, without integrating multiple factors into a unified congestion detection framework. Furthermore, optical flow and motion-based methods may struggle under heavy occlusion, complex perspectives, and dense mixed traffic conditions.

The third category involves detection and multi-object tracking approaches, which enable a more comprehensive understanding of traffic by capturing both spatial and temporal characteristics. Shen et al. (2025) propose a real-time surveillance system combining YOLOv8 and ByteTrack for accurate detection and tracking of traffic participants, demonstrating strong performance with low computational cost [20]. In a similar direction, the study “Passable: An Intelligent Traffic Light System with Integrated Deep Learning and Computer Vision” (2025) introduces an adaptive traffic signal control system that adjusts signal timing based on real-time traffic density [21]. Another work, “Real-Time Traffic Control and Monitoring” (2023), employs EfficientDet and TensorFlow Lite to estimate traffic “bulkiness” from live video

streams for control purposes [22]. Additionally, "Smart Traffic Management of Vehicles Using Faster R-CNN-Based Deep Learning Method" (2024) explores the use of two-stage detectors for vehicle detection in dense and occluded environments, achieving competitive detection accuracy [23]. While these approaches provide more interpretable outputs such as vehicle count, flow, and spatial distribution, many of them focus primarily on surveillance or simplified indicators like density, without integrating key factors such as speed and temporal persistence into a unified congestion model.

Overall, despite significant progress across these three research directions, several limitations remain. Image-level classification methods lack interpretability, motion-based approaches often fail in complex urban scenarios, and detection–tracking frameworks do not fully exploit multi-factor traffic indicators. In particular, there is still a lack of robust and adaptive models capable of simultaneously incorporating vehicle density, speed, and temporal duration to accurately identify congestion in mixed traffic environments, such as those commonly observed in developing countries. These challenges motivate the development of more comprehensive frameworks that combine detection, tracking, and multi-factor analysis for real-time traffic congestion detection.

3. METHOD

This section presents the proposed framework for real-time traffic congestion detection at urban intersections. The system is designed to process traffic video streams and infer congestion status by combining vehicle detection, multi-object tracking, and traffic metric analysis. The overall pipeline consists of four main stages: input acquisition, vehicle detection, object tracking, and congestion decision-making. In particular, the proposed method emphasizes region-based analysis and a mixed-traffic representation using MEU, allowing the framework to better adapt to urban traffic environments where different vehicle types occupy different amounts of road space.

3.1. System pipeline

The overall system pipeline, as illustrated in Fig. 1, consists of a sequence of processing stages, including input video acquisition, vehicle detection, multi-object tracking, traffic metric computation, and congestion decision.

At the first stage, traffic video streams are captured from surveillance cameras installed at urban intersections. These video frames serve as the primary input for the system. Each frame is then processed by a vehicle detection model, specifically YOLOv8, to identify and classify different types of traffic participants such as motorcycles, cars, buses, and trucks.

In the next stage, a multi-object tracking algorithm, namely ByteTrack, is employed to associate detected vehicles across consecutive frames. This process assigns a unique identity to each vehicle and enables the reconstruction of its trajectory over time. As a result, the system can maintain temporal consistency and avoid duplicate counting of vehicles.

Following the tracking stage, the system performs traffic analysis within a predefined ROI. Only vehicles whose positions fall inside the ROI are considered for further processing. Based on these tracked vehicles, several traffic metrics are computed, including vehicle density, average velocity, and temporal persistence. To better reflect mixed traffic conditions, different vehicle types are converted into a unified representation using MEU, allowing the system to account for their varying impact on road occupancy.

Finally, the computed traffic indicators are integrated through a rule-based decision mechanism to determine the congestion status. Specifically, congestion is identified when the density exceeds a predefined threshold, the average velocity falls below a certain limit, and these conditions persist over a specified duration. The output of the system is a binary or categorical congestion state, indicating whether the traffic condition is normal or congested.

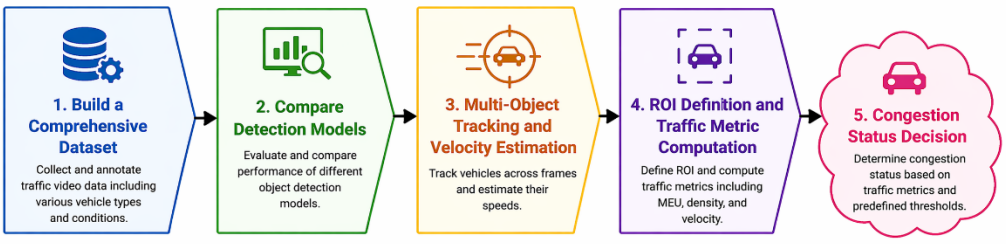


Figure 1. Proposed system pipeline for traffic congestion detection.

The proposed pipeline has two key characteristics. First, it performs localized analysis by focusing on vehicles within user-defined ROIs, making it suitable for intersection-level monitoring rather than full-frame analysis. Second, it is designed to handle mixed traffic environments, where heterogeneous vehicle types coexist. By incorporating MEU-based normalization, the system provides a more realistic representation of traffic conditions compared to simple vehicle counting methods.

3.2. Vehicle detection

The vehicle detection stage is responsible for locating and classifying traffic participants in each frame. In this study, YOLOv8 is adopted as the detector due to its favorable balance between detection accuracy and real-time performance. Compared with lighter variants, YOLOv8 provides stronger feature extraction capability, which is beneficial in dense traffic scenes with overlapping or partially occluded vehicles, while still maintaining sufficiently low latency for real-time deployment compared to heavier detectors. Given an input frame I_t at time t, the detector outputs a set of bounding boxes, denoted as D_t, as defined in (1).

$$D_t = \{(b_i, c_i, s_i)\}_{i=1}^{N_t} \tag{1}$$

where: $b_i = (x_i, y_i, w_i, h_i)$ denotes the bounding box of the i-th detected vehicle, c_i is the class label, s_i is the confidence score, N_t is the number of detections in frame t.

The detector considers five main vehicle categories: motorcycle, bicycle, car, truck, and bus. These categories were selected because they are common in urban traffic and sufficiently represent the diversity of mixed traffic flow. The use of class-specific outputs is essential for the subsequent MEU conversion, where different vehicle types contribute differently to congestion estimation.

To reduce false positives, only detections with confidence scores above a predefined threshold are retained. The filtered detections are then passed to the tracking stage.

3.3. Multi-object tracking

After detection, the system performs multi-object tracking to preserve the identities of vehicles across frames and recover their trajectories over time. This study employs ByteTrack because of its strong performance in crowded scenes and its ability to recover valid objects even when their confidence scores are relatively low. Specifically, the detection results in frame t, denoted as D_t, are divided into two groups: high-confidence detections D_t^{high} and low-confidence detections D_t^{low}.

ByteTrack performs data association in two stages, where existing tracks are first matched with high-confidence detections using Intersection over Union (IoU)-based similarity, followed by a second stage in which unmatched tracks are associated with low-confidence detections to recover vehicles that may be partially occluded or weakly detected. The matching cost between a track T_j and a detection D_i is defined as in (2).

$$C_{ij} = 1 - IoU(T_j, D_i) \tag{2}$$

where: $IoU(T_j, D_i) = \frac{Area(T_j \cap D_i)}{Area(T_j \cup D_i)}$

The Hungarian algorithm is then applied to minimize the total association cost and produce the optimal matching between current detections and existing tracks, resulting in a set of tracked objects, denoted as T_t, as defined in (3).

$$T_t = \{(id_k, b_k, c_k)\}_{k=1}^{M_t} \tag{3}$$

where: id_k is the unique ID of the tracked vehicle, b_k is its current bounding box, c_k is its class label, M_t is the number of active tracks.

Tracking plays a crucial role in the proposed system for three reasons. First, it prevents duplicate counting by ensuring that the same vehicle is not counted multiple times across frames. Second, it provides temporal continuity, which is necessary for estimating vehicle speed. Third, it improves system robustness in dense traffic scenes with occlusion and overlapping vehicles.

3.4. Congestion metric

The congestion decision module is the core contribution of the proposed framework. Instead of treating congestion as a pure image-level classification problem, the system estimates congestion from interpretable traffic indicators derived from tracked vehicles inside an ROI. Four key components are used: ROI filtering, MEU conversion, density estimation, and velocity estimation.

3.4.1. Region of interest

A ROI is defined as a polygonal area corresponding to a traffic lane, waiting zone, or intersection segment to be monitored. Let the centroid of vehicle *k* at frame *t* be denoted as $p_k^t = (x_k^t, y_k^t)$. A vehicle is included in the congestion analysis if its centroid satisfies $p_k^t \in ROI$.

This ROI-based filtering allows the system to focus on specific traffic regions rather than the entire frame, improving both computational efficiency and interpretability.

3.4.2. Motorcycle equivalent unit

In mixed traffic environments, simply counting vehicles is insufficient because different vehicle types occupy different amounts of space and influence traffic flow differently. To address this issue, the proposed framework converts each detected vehicle into a MEU.

Let n_c denote the number of vehicles of class *c* inside the ROI, and let α_c be the MEU coefficient assigned to class *c*. Then, the total equivalent traffic load is defined as in (4).

$$MEU_{total} = \sum_{c=1}^{C} \alpha_c * n_c \tag{4}$$

where *C* is the number of vehicle classes.

In this study, the MEU coefficients are assigned as follows: motorcycle: 1.0, bicycle: 1.0, car: 3.4, truck: 6.0, and bus: 10.5. These coefficients reflect the relative road occupancy and traffic influence of each vehicle type compared to a motorcycle [3].

3.4.3. Density estimation

Traffic density is estimated based on the total MEU load inside the ROI. Let A_{ROI} denote the effective area of the ROI. Then the density can be defined as in (5).

$$\rho_t = \frac{MEU_{total}(t)}{A_{ROI}} \tag{5}$$

where: ρ_t is the traffic density at time *t*, $MEU_{total}(t)$ is the total equivalent vehicle load at time *t*.

In practice, if precise geometric calibration is not available, the density can be approximated directly by the total equivalent number of vehicles inside the ROI as (6).

$$\rho_t' = MEU_{total}(t) \tag{6}$$

This approximation is often sufficient when the ROI is fixed for a given camera setting.

3.4.4. Velocity estimation

Vehicle speed is estimated from the displacement of tracked centroids across frames. Let the centroid of vehicle *k* at frames *t*-1 and *t* be p_k^{t-1} and p_k^t, respectively. The displacement in image coordinates is defined as in (7).

$$\Delta d_k^t = \|p_k^t - p_k^{t-1}\| \tag{7}$$

If the video frame rate is *FPS* and a spatial calibration factor β is available to convert pixel distance into real-world distance, the speed of vehicle *k* is computed as (8). The resulting speed can then be converted into km/h by applying an appropriate unit conversion factor.

$$v_k^t = \Delta d_k^t * FPS * \beta \tag{8}$$

The average speed of all tracked vehicles inside the ROI at time *t* is computed as in (9).

$$\overline{v}_t = \frac{1}{M_t^{ROI}} \sum_{k=1}^{M_t^{ROI}} v_k^t \quad (9)$$

where M_t^{ROI} is the number of tracked vehicles inside the ROI.

3.4.5. Congestion decision logic

The congestion thresholds used in this study were determined empirically based on real-world traffic observations and practical traffic monitoring criteria commonly used in Vietnam. In particular, the speed threshold of 5 km/h was selected because urban traffic conditions with average vehicle speeds below this level are generally considered congested in practical traffic monitoring systems. The density threshold of 50 MEU-equivalent vehicles inside the ROI was determined through repeated observations on traffic videos collected at major intersections in Ho Chi Minh City, where traffic conditions above this threshold consistently corresponded to visually congested states. In addition, a temporal persistence threshold of 2 minutes was introduced to reduce false congestion alarms caused by temporary stopping behavior during traffic-light cycles or short-term fluctuations in vehicle flow. Although fixed thresholds were adopted in this study for interpretability and practical deployment, future work may investigate adaptive threshold estimation methods based on historical traffic patterns and environmental conditions.

Algorithm 1. Real-Time traffic congestion detection

Input:

Video stream V

Predefined ROI

Detection model YOLOv8

Tracking model ByteTrack

Density threshold τ_ρ

Speed threshold τ_v

Time threshold τ_T

Output: Congestion status for each time interval

```
Initialize tracker
For each frame I_t in video stream V do
    Detect vehicles using YOLOv8
    Filter detections by confidence threshold
    Track detected vehicles using ByteTrack
    Select tracked vehicles whose centroids lie inside ROI
    Count vehicles by class inside ROI
    Convert vehicle counts into MEU_total
    Estimate speed of each tracked vehicle
  Compute average speed v_t inside ROI
  If (MEU_total > τ_ρ) and (v_t < τ_v) continuously for τ_T then
      Set congestion status = 1
  Else
      Set congestion status = 0
  End if
End for
Return congestion status
```

The system determines the congestion status by jointly considering traffic density, average speed, and persistence duration. A traffic region is considered to be in a congested state if the following conditions are simultaneously satisfied, as defined in (10).

$$\begin{aligned} \rho_t' &> \tau_\rho \\ \overline{v}_t &< \tau_v \\ \Delta t_{persist} &> \tau_T \end{aligned} \tag{10}$$

where: τ_ρ is the density threshold, τ_v is the speed threshold, τ_T is the temporal persistence threshold. In this study, the default thresholds are: $\tau_\rho = 50$ MEU-equivalent vehicles in ROI, $\tau_v = 5\ km/h$, $\tau_T = 2\ minutes$.

Thus, the congestion decision function can be formulated as in (11).

$$Congestion_t = \begin{cases} 1, & if\ \rho_t' > \tau_\rho, \overline{v}_t < \tau_v, and\ \Delta t_{persist} > \tau_T \\ 0, & otherwise \end{cases} \tag{11}$$

This rule-based formulation makes the system interpretable and practical. Instead of predicting congestion as a black-box label, the model explains congestion through physically meaningful traffic indicators. This is especially useful in intelligent traffic systems, where the output may support downstream tasks such as adaptive traffic signal control.

The complete procedure for real-time congestion detection is summarized in Algorithm 1. This algorithm formalizes the proposed framework by integrating vehicle detection, tracking, ROI-based filtering, MEU computation, and multi-factor decision logic into a unified process. In particular, the congestion state is determined by jointly evaluating density, average velocity, and temporal persistence over a predefined time window.

4. EXPERIMENTAL RESULTS AND EVALUATION

4.1. Data Collection

The dataset used in this study was constructed from three primary sources: the AI Challenge HCMC 2020 – Vehicle Counting Dataset, the COCO dataset (specifically selected classes such as buses and bicycles), and a custom dataset collected by the research team.

For the custom data, images and video sequences were captured at pedestrian overpasses in Ho Chi Minh City. The cameras were positioned at elevated viewpoints and oriented downward toward intersections, simulating the perspective of real-world traffic surveillance systems. This setup ensures that the collected data closely reflects practical deployment conditions in urban traffic monitoring.

The raw data were manually annotated using Roboflow, where bounding boxes were created and class labels were assigned to each detected object. In total, the dataset consists of 1,268 labeled images with 4,933 annotated objects, which were used for training the detection model. Table 1 summarizes the distribution of object classes in the dataset.

Table 1. Dataset statistics

No.	Class	Number of objects
1	Car	1,025
2	Bicycle	674
3	Bus	567
4	Motorcycle	1,921
5	Truck	746

Before training and evaluation, the dataset was preprocessed through several steps. First, images were resized to a standard resolution of 640×640 pixels. Pixel values were normalized to the range [0, 1] to stabilize model training. In addition, data augmentation techniques such as random cropping and perspective transformations were applied to increase data diversity and improve model robustness under varying real-world conditions.

Finally, the dataset was divided into training, validation, and testing subsets with a ratio of 70%, 10%, and 20%, respectively.

4.2. Model selection and detection performance

4.2.1 Comparison of object detection models

In the initial stage of the experiments, several object detection models, including SSDLite, Faster R-CNN, and YOLO-based models, were evaluated on the same traffic dataset to assess their effectiveness in detecting vehicles in urban traffic conditions in Vietnam.

As shown in Table 2, YOLO-based models significantly outperform traditional detection approaches such as Faster R-CNN and SSDLite. SSDLite exhibits the lowest performance, with a Precision of 0.24 and Recall of 0.37, making it unsuitable for real-world traffic monitoring. Faster R-CNN achieves a relatively high mAP@0.5 of 0.85, but its low Recall (0.27) indicates that many vehicles are missed, which limits its applicability in traffic analysis.

Table 2. Performance comparison of object detection models

Model	Precision	Recall	mAP@0.5	mAP@0.5:0.95
Faster R-CNN	0.57	0.27	0.85	0.49
SSDLite	0.24	0.37	0.504	0.32
YOLOv5	0.85	0.85	0.89	0.53
YOLOv7	0.862	0.861	0.915	0.596
YOLOv8	0.882	0.845	0.911	0.597

Among the YOLO-based models, YOLOv5 demonstrates strong performance with balanced Precision and Recall values (both 0.85), serving as a solid baseline. However, more recent models, including YOLOv7 and YOLOv8, provide further improvements. YOLOv7 achieves higher Precision (0.862) and Recall (0.861), along with improved mAP values, due to its enhanced feature aggregation architecture. YOLOv8 achieves the highest overall Precision (0.882) and competitive mAP performance, demonstrating the effectiveness of its anchor-free design and improved label assignment strategy.

Overall, the results indicate that YOLO-based detectors are more suitable for traffic applications. In particular, YOLOv8 provides the best trade-off between detection accuracy and robustness, and is therefore selected as the backbone model for the proposed system.

4.2.2. Detailed evaluation of YOLOv8

After selecting YOLOv8 as the detection model, a detailed evaluation is conducted to analyze its performance on the traffic dataset.

Table 4. Training Results of the YOLOv8 Model

Class	Precision	Recall	mAP@0.5	mAP@0.5:0.95
All	0.882	0.845	0.911	0.597
Car	0.856	0.865	0.927	0.585
Bicycle	0.942	0.852	0.926	0.618
Bus	0.911	0.919	0.951	0.729
Motorcycle	0.818	0.787	0.861	0.449
Truck	0.88	0.804	0.89	0.604

As shown in Table 4, YOLOv8 achieves an overall Precision of 0.882, Recall of 0.845, mAP@0.5 of 0.911, and mAP@0.5:0.95 of 0.597, demonstrating strong detection performance and robustness.

Among the five vehicle classes, buses and cars achieve the highest performance with mAP@0.5 values of 0.951 and 0.927, respectively. In contrast, motorcycles and trucks exhibit lower performance due to their smaller size, high variability, and frequent occlusion in dense traffic environments.

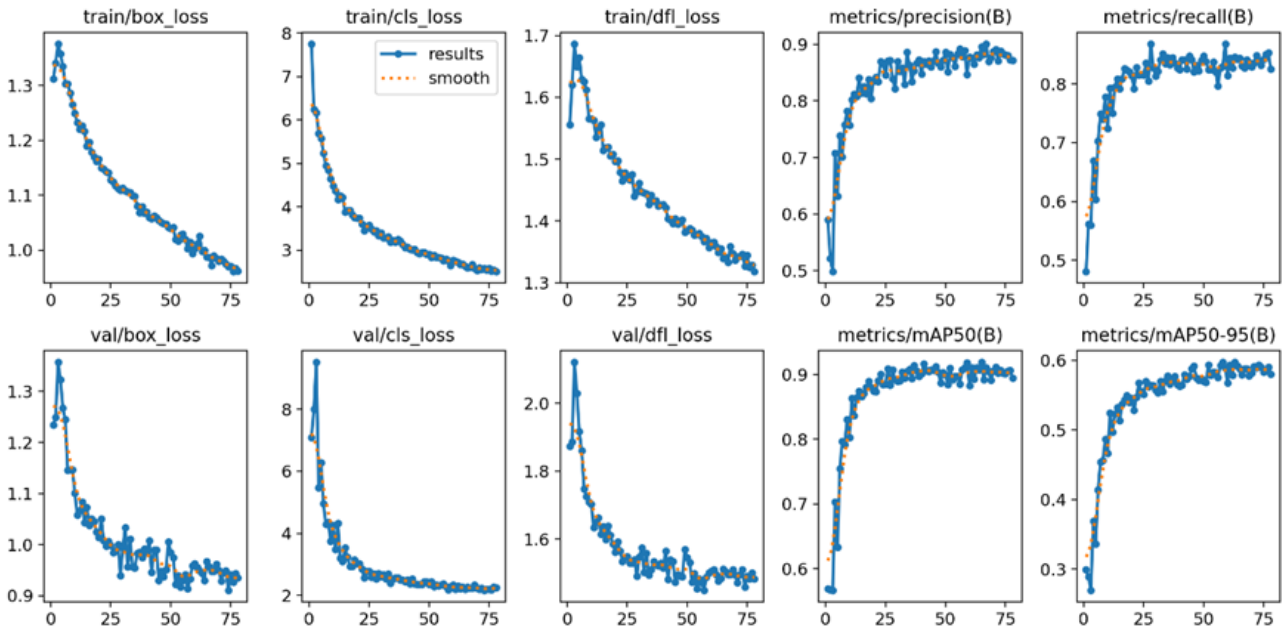


Figure 2. Training results of YOLOv8: loss and evaluation metrics over epochs

The training process is illustrated in Figure 2, where the loss components (box_loss, cls_loss, and dfl_loss) decrease steadily over epochs, indicating stable learning. The validation losses follow a similar trend without significant divergence, suggesting that the model does not suffer from overfitting.

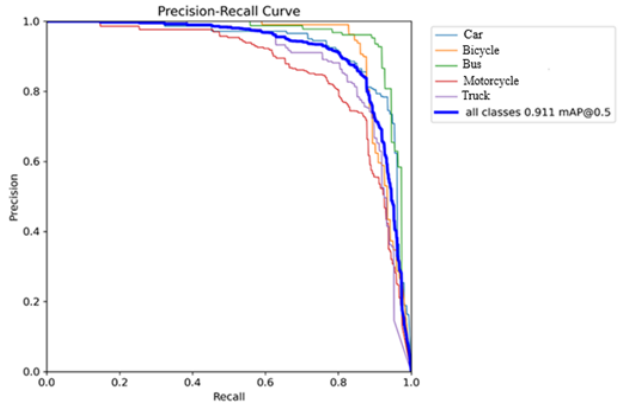


Figure 3. Precision–recall curves of YOLOv8 for each vehicle class

Regarding evaluation metrics, both precision and recall increase rapidly during the early training phase and stabilize afterward. The mAP@0.5 approaches 0.9, while mAP@0.5:0.95 reaches approximately 0.6, which is reasonable given the diversity of traffic conditions and object appearances.

The Precision–Recall curves in Figure 3 show that the model maintains high precision across most recall levels for the majority of vehicle classes. The bus class achieves the best performance, while motorcycles show lower performance due to their small size and high occlusion rate.

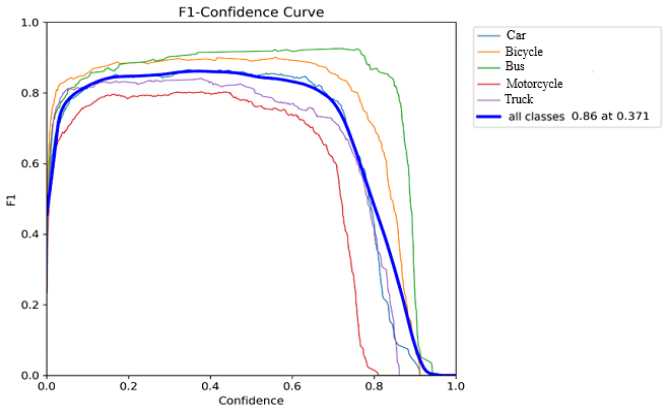


Figure 4. F1–confidence curves of YOLOv8 for each vehicle class

The F1–confidence curve (Figure 4) indicates that the optimal confidence threshold is around 0.37, where the F1-score reaches its maximum value of approximately 0.86. This reflects a good balance between precision and recall.

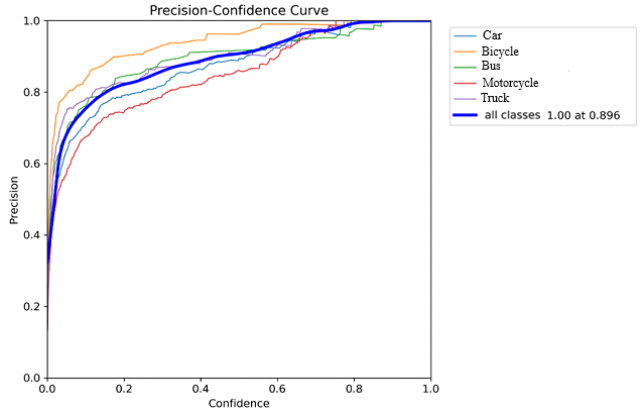


Figure 5. Class-wise precision–confidence curves of the YOLOv8 model

The Precision–confidence curve (Figure 5) demonstrates that precision increases significantly with higher confidence thresholds, reaching near-perfect precision at high confidence levels. Meanwhile, the Recall–confidence curve (Figure 6) shows that recall remains high at low confidence thresholds but decreases as the threshold increases, particularly for challenging classes such as motorcycles.

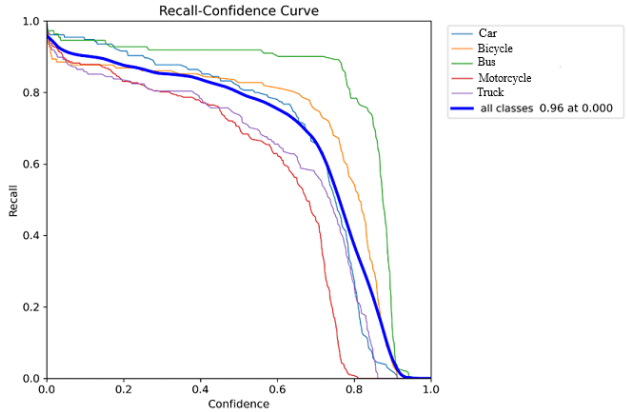


Figure 6. Class-wise recall–confidence curves of the YOLOv8 model

Overall, the results confirm that YOLOv8 is well-suited for vehicle detection in urban traffic environments, providing a good trade-off between detection accuracy and robustness. However, performance on small or heavily occluded objects still presents challenges and may be improved through further data augmentation or model refinement.

4.3 Congestion detection performance

To evaluate the effectiveness of the proposed congestion detection method, experiments were conducted on real-world traffic video sequences. The system determines traffic congestion based on vehicle density within the ROI and average vehicle speed.

In congested scenarios, the vehicle density exceeds the predefined threshold (τ_ρ), while the average vehicle speed remains below the speed threshold (τ_v) for a continuous duration (τ_T). Conversely, in non-congested conditions, vehicles exhibit higher speeds and lower density values.

Figure 7 illustrates qualitative results of the proposed system. When traffic congestion occurs, the system correctly identifies the congested region and labels it accordingly. In contrast, under normal traffic conditions, the system maintains a non-congested status.

Although ground truth annotations for congestion are not available, the experimental observations confirm that the system produces consistent and reliable results across different traffic conditions. The combination of spatial density and motion-based metrics allows the system to detect congestion even in complex urban environments.

4.4 Qualitative evaluation under real-world scenarios

To further validate the effectiveness of the proposed system, qualitative experiments were conducted on real-world traffic videos collected by the research team. These videos were not used during training or validation, ensuring an unbiased evaluation of the system performance.

The dataset consists of video sequences captured from pedestrian overpasses at urban intersections in Ho Chi Minh City, featuring typical traffic conditions with 2–4 lanes. All experiments were conducted under natural daylight conditions to reflect realistic deployment scenarios.

Two representative traffic conditions were selected for evaluation:

Scenario 1: Congested traffic under favorable lighting conditions

This scenario evaluates the ability of the system to detect congestion when traffic density is high and vehicle speeds are low. The results show that the system successfully detects and tracks vehicles while correctly identifying the congested state within the region of interest.

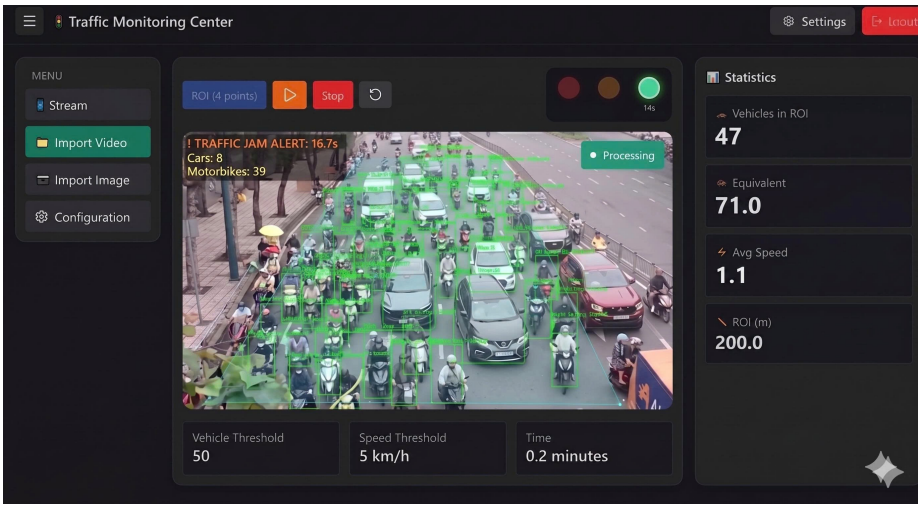


Figure 7. Congested traffic scenario: visualization of vehicle detection and tracking results within the defined ROI. The system identifies multiple vehicles with assigned tracking IDs, while high vehicle density and low average speed lead to a congestion alert.

Scenario 2: Normal traffic flow under favorable lighting conditions

This scenario evaluates the system under normal traffic conditions, where vehicles are moving at higher speeds, even when traffic volume is relatively high. The system is able to correctly distinguish this situation from congestion, demonstrating robustness against false positive detection.

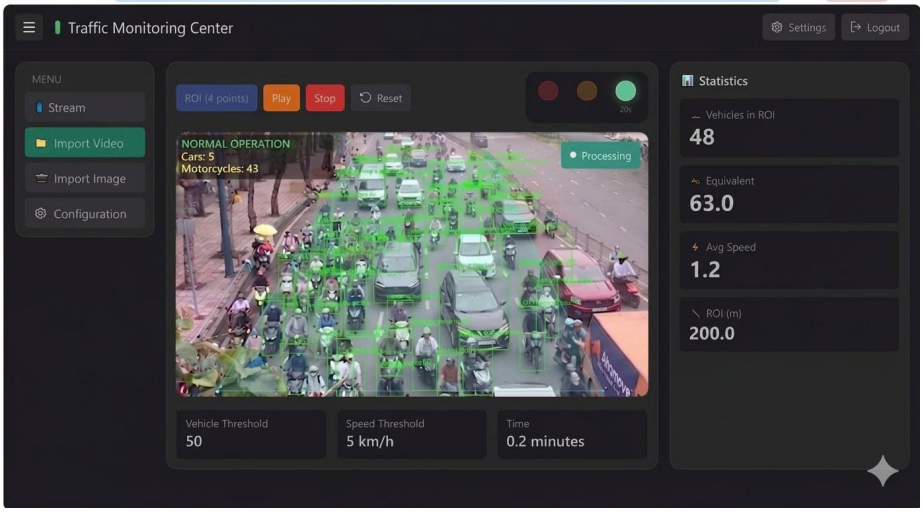


Figure 8. Normal traffic scenario: visualization of vehicle detection and tracking results under non-congested conditions. Although multiple vehicles are present, higher movement speed and lower density prevent the system from triggering a congestion state.

These experiments demonstrate that the proposed system is capable of reliably distinguishing between congested and non-congested traffic states in real-world environments. The combination of density and motion-based analysis enables the system to provide stable and interpretable results under different traffic conditions.

5. CONCLUSION

This paper presents a real-time traffic congestion detection framework for mixed traffic environments based on deep learning and multi-object tracking. By integrating YOLOv8 for vehicle detection and ByteTrack for multi-object tracking, the proposed system is capable of analyzing traffic conditions within a predefined region of interest (ROI) while maintaining temporal consistency of tracked vehicles. In addition, the use of Motorcycle Equivalent Unit (MEU)-based traffic representation enables the framework to better reflect heterogeneous traffic characteristics commonly observed in urban areas such as Ho Chi Minh City. The congestion decision mechanism combines vehicle density, average speed, and temporal persistence, providing a more interpretable and practical approach for congestion detection compared with image-level classification methods.

Experimental results demonstrate that the proposed framework achieves competitive vehicle detection performance and is capable of reliably distinguishing between congested and non-congested traffic conditions in real-world scenarios. The integration of spatial density analysis and motion-based information improves the robustness of the system under complex urban traffic environments. Furthermore, the ROI-based analysis allows localized monitoring and reduces the influence of irrelevant traffic regions, making the framework suitable for intelligent traffic monitoring applications and potential adaptive traffic signal control systems.

Despite the promising results, the proposed system still has several limitations. First, the experiments were mainly conducted under daytime and favorable weather conditions; therefore, detection and tracking performance may decrease at nighttime or during heavy rain due to reduced visibility and image quality. Second, camera vibration and unstable viewpoints may affect tracking consistency and speed estimation accuracy. Third, the current framework assumes a relatively fixed camera angle and ROI configuration, meaning that significant viewpoint changes may require recalibration of the ROI and spatial conversion parameters. Finally, extremely dense motorcycle traffic with severe occlusion remains challenging for accurate vehicle detection and tracking.

Future work will focus on improving system robustness under adverse environmental conditions, incorporating adaptive threshold and calibration strategies, and extending the framework for large-scale real-world deployment in intelligent transportation systems.

REFERENCES

[1] T. Litman, "The costs of traffic congestion," Victoria Transport Policy Institute, 2020.
[2] World Bank, "Urban transport and congestion in developing cities," 2019.
[3] Q. Cao and K. Sano, "Evaluation of motorcycle equivalent units in mixed traffic flow," Journal of Transportation Engineering, vol. 138, no. 9, pp. 1146–1155, 2012.
[4] S. Sivaraman and M. M. Trivedi, "Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis," IEEE Trans. Intell. Transp. Syst., vol. 14, no. 4, pp. 1773–1795, 2013.
[5] J. Redmon et al., "You only look once: Unified, real-time object detection," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016.
[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, 2017.
[7] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in Proc. IEEE Int. Conf. Image Process. (ICIP), 2016, pp. 3464–3468.
[8] Y. Zhang et al., "ByteTrack: Multi-object tracking by associating every detection box," in Proc. Eur. Conf. Comput. Vis. (ECCV), 2022.
[9] N. T. Luong, "Traffic congestion detection from camera images using CNN," Bachelor's thesis, 2021.
[10] B. K. P. Horn and B. G. Schunck, "Determining optical flow," Artificial Intelligence, vol. 17, no. 1–3, pp. 185–203, 1981.
[11] Ultralytics, "YOLOv8 documentation," 2023. [Online]. Available: https://docs.ultralytics.com
[12] A. Kumar, S. Sharma, and R. Gupta, "MobileNetV2 with spatial attention module for traffic congestion recognition," Expert Syst. Appl., vol. 235, 2024.
[13] Y. Jiang and X. Feng, "Traffic congestion detection using a small parallel residual convolutional neural network," Sci. Rep., vol. 15, no. 1, 2025.
[14] H. Liu, Z. Wang, and J. Chen, "Traffic congestion recognition based on convolutional neural networks in different scenarios," Eng. Appl. Artif. Intell., vol. 130, 2025.
[15] M. Patel, K. Singh, and P. Verma, "Traffic congestion detection from camera images using deep learning approach," in Proc. Int. Conf. Intell. Syst. Appl., Springer, 2024, pp. 325–336.
[16] M. R. R. Siddiqui, A. Khan, and S. Lee, "FARSEC: A reproducible framework for real-time vehicle speed estimation from traffic cameras," arXiv preprint arXiv:2309.14468, 2023.
[17] M. Ashraf, M. A. Khan, S. U. Rehman, and T. Saba, "HVD-Net: A hybrid vehicle detection network for vision-based vehicle tracking and speed estimation," J. King Saud Univ. Comput. Inf. Sci., vol. 35, no. 7, pp. 101–113, 2023.
[18] M. A. Kamil, M. H. Habaebi, and M. R. Islam, "Robust vehicle speed estimation based on vision sensor using YOLOv5 and DeepSORT," in Proc. Int. Conf. Comput. Vis. Image Process., Springer, 2023, pp. 45–56.
[19] O. Azimjonov, J. Park, and K. Lee, "A vision-based real-time traffic flow monitoring system for road intersections," Multimed. Tools Appl., vol. 82, pp. 15021–15045, 2023.
[20] Y. Shen, H. Zhang, and L. Wang, "Enhancing traffic surveillance through YOLOv8 and ByteTrack for real-time object detection and tracking," in Proc. Int. Conf. Intell. Transp. Syst., Springer, 2025, pp. 112–123.
[21] M. Alharbi, S. Khan, and A. Alzahrani, "Passable: An intelligent traffic light system with integrated deep learning and computer vision," Sensors, vol. 25, no. 18, p. 5760, 2025.
[22] R. K. Singh, P. Kumar, and A. Verma, "Real-time traffic control and monitoring using deep learning and edge computing," Results Eng., vol. 18, p. 101234, 2023.
[23] S. Patel, R. Mehta, and K. Shah, "Smart traffic management of vehicles using Faster R-CNN based deep learning method," Sci. Rep., vol. 14, no. 1, p. 60596, 2024.