EAI Endorsed Transactions
on Internet of Things
Research Article EAI.EU

Multi-View Vehicle Detection and Tracking for Smart City Traffic Monitoring

Thang C. Vu[1], Dung T. Nguyen[1], Long Q. Dinh[1], Mui D. Nguyen[2], Minh T. Nguyen[2,*]

[1] Thai Nguyen University of Information and Communication Technology, Thai Nguyen, Viet Nam
[2] Thai Nguyen University of Technology, Thai Nguyen University, Thai Nguyen, Viet Nam

Abstract

Urban traffic monitoring plays a crucial role in intelligent transportation systems. The development of surveillance camera networks has generated a large amount of image and video data that can be exploited for traffic flow detection, tracking, and analysis tasks. However, detecting and tracking vehicles from fixed traffic surveillance cameras still faces many challenges. The main challenges include obscured objects, small target size, and high traffic density. This study presents a deep learning-based traffic monitoring framework for detecting and tracking multiple objects in urban traffic monitoring systems. The proposed framework integrates YOLOv11 for vehicle detection and DeepSORT with a Kalman filter-based state estimation method for tracking multiple objects. In addition, the SAHI technique is integrated to investigate its ability to support the detection of small objects in traffic data. The research framework was evaluated using a dataset collected from traffic cameras in Thai Nguyen, Vietnam. Numerous test scenarios were conducted with varying traffic densities, observation distances, and camera viewing angles. Experimental results showed that the YOLOv11 configuration combined with DeepSORT achieved a processing speed of approximately 10.1 FPS; for object detection tasks, the model achieved an mAP@0.5 of 0.66. Simultaneously, experimental results show that the proposed framework can maintain vehicle detection and tracking across consecutive frames under varying observation conditions. In addition, the integration of SAHI techniques recorded an improvement in detecting small objects, with mAP@0.5 increasing from 0.66 to 0.70 and AP_S increasing from 0.29 to 0.40. The results obtained demonstrate the potential applicability of the proposed framework to traffic detection, tracking, and monitoring problems in urban environments.

Keywords: Deep Learning, Intelligent transportation systems, YOLOv11, Vehicle Detection, Multi-Object Tracking, SAHI

Received on 29 March 2026, accepted on 20 June 2026, published on 30 June 2026

doi: 10.4108/12412

1. Introduction

In recent years, rapid urbanization has significantly increased pressure on transportation systems in cities, especially in developing urban areas [1, 2]. In this context, urban traffic monitoring plays a crucial role in supporting the development of Smart City applications, aiming for efficient and sustainable traffic management [3, 4]. The expansion of surveillance camera systems at intersections and along major roads has generated a large and cost-effective data source, contributing to improved data-driven traffic analysis and management. However, effectively exploiting data from urban traffic camera systems still faces many challenges. In practice, the images obtained are often affected by the phenomenon of obstruction between vehicles, changes in environmental conditions and high traffic density. In addition, the diversity of vehicle types also increases the complexity of the object detection and tracking problem. Especially in the context of urban traffic in Vietnam, the large proportion of motorbikes and small

*Corresponding author. Email: nguyentuanminh@tnut.edu.vn

vehicles makes object detection and tracking from surveillance cameras more difficult. Therefore, traditional object detection methods often have limitations in accuracy and stability when applied in real traffic scenarios [5].

In recent times, the development of artificial intelligence (AI) has made significant progress [6–8], particularly in deep learning (DL)-based models for the field of computer vision [9, 10]. Many computer vision techniques have been proposed to serve the problem of vehicle detection and traffic analysis. Several previous studies have used traditional image processing techniques to identify moving vehicles in video sequences [11, 12]. However, these methods often encounter many limitations in complex urban traffic environments due to the influence of factors such as lighting conditions, occlusion and background noise, etc. In recent years, along with the rapid development of DL techniques, convolutional neural network (CNN) models have shown significant effectiveness in object detection, classification and tracking problems in images and videos [13-15]. For example, the YOLOv3 model was combined with k-means clustering techniques [13] to improve accuracy in object recognition and tracking. A proposed architecture [14] based on YOLOv9 integrates attention mechanisms and multi-stage convolutional layers to improve the ability to detect and track small vehicles under conditions of occlusion or complex backgrounds. Research in [16] proposed a method to support urban traffic monitoring and control based on multi-scale CNNs to improve image processing capabilities under adverse weather conditions. The author in [17] proposes a vehicle segmentation technique based on adaptive amplification to reduce the effects of shadows and different lighting conditions. In addition, many studies have proposed combining YOLO models with tracking algorithms or intelligent processing components to enhance object detection capabilities and support intelligent traffic applications [18, 19]. For example, an intelligent traffic monitoring system based on YOLOv5 has been developed to detect road surface anomalies and assist drivers [18]. The results have demonstrated the potential of machine learning and object detection techniques in enhancing traffic monitoring applications.

Despite these advances, several challenges remain. Specifically, many studies focus primarily on single-camera scenarios and do not adequately address the challenges of multi-camera environments with different viewing angles. Furthermore, detecting small vehicles at long distances remains a critical issue, especially in very high-positioned camera setups for comprehensive viewing angles. Although several solutions have been proposed, including inference techniques supported by image splitting, where large images are divided into small overlapping arrays to improve the ability to identify small targets [20, 21] several other solutions have been mentioned based on generating adversarial networks [22], based on attention mechanisms [23, 24], based on feature combination from multiple levels [25] or based on exploiting contextual information [26], etc. However, these methods often increase model complexity and computational costs, or their integration into traffic monitoring systems remains limited.

This paper proposes a vehicle detection and monitoring framework for urban traffic monitoring applications. The system uses video data collected from surveillance cameras deployed at various locations and heights in an urban environment, combined with a centralized processing architecture to support data integration. This framework combines the YOLOv11 model with the DeepSORT algorithm, integrating estimation methods such as the Kalman filter for target tracking. In addition, the Slicing-Aided Hyper Inference (SAHI) technique was integrated with YOLOv11 to investigate the ability to detect small objects at long distances in a traffic monitoring context.

Unlike many studies that focus primarily on model improvement or evaluation on single-camera scenarios, this research aims to build and evaluate a deployment-oriented system framework in the context of real-world urban traffic. Furthermore, the study examines the potential of leveraging existing surveillance camera infrastructure in a multi-camera environment with non-uniform viewing angles.

Experimental results of the system framework show that the number of small-sized vehicles detected tends to increase, while the system maintains relative stability of the tracking process under the surveyed traffic conditions. Specifically, the YOLOv11 model combined with the DeepSORT algorithm maintains consistent object detection and tracking across consecutive frames. The results obtained demonstrate the potential application of this framework in traffic flow and density analysis, helping to determine the state of the traffic system in specific situations. Furthermore, the integration of SAHI is considered as an approach to support the detection of small vehicles at long distances. This contributes to improving the detection of missed objects and demonstrates the feasibility of the proposed system framework in high-density traffic scenarios with diverse viewing angles.

The main contributions of this paper include:

- This study proposes an integrated urban traffic monitoring framework, combining the YOLOv11 model, the DeepSORT algorithm, and SAHI techniques to support vehicle detection and tracking tasks in urban traffic environments through a unified processing procedure, aiming for practical implementation.
- This study examines the applicability of the proposed framework to data collected from multiple surveillance cameras with varying installation locations and viewing angles. The test scenarios reflect several common conditions in real-world traffic surveillance systems and demonstrate the potential for leveraging data from existing surveillance camera infrastructure.
- This study examines the potential for improving the detection of small vehicles at long distances by

combining YOLOv11 with SAHI techniques on wide-area surveillance images.
- This paper presents experimental results on real-world traffic data collected in Thai Nguyen City, Vietnam, thereby illustrating the applicability of the proposed framework in vehicle monitoring and tracking problems.

The rest of this paper is organized as follows. Section 2 summarizes the theoretical frameworks and related methods used in the research. Section 3 presents the proposed multi-view traffic monitoring framework. This section focuses on the system architecture, the multi-object detection and tracking process based on YOLOv11 and DeepSORT, and the integration of SAHI techniques as a supporting component for detecting small or distant objects in traffic monitoring images. Section 4 describes the dataset, evaluation protocol, and experimental results on real-world traffic data collected in Thai Nguyen, Vietnam. Finally, conclusions and future research directions are provided in Section 5.

2. Theoretical Background and Related Methods

2.1. State estimation using the Kalman Filter

The linear estimation problem is implemented based on Kalman Filter (KF). Equations (1) and (2) describe a set of states and measurements [27].

$$x_{k+1} = Ax_k + Bu_k + \mathrm{w}_k, \quad (1)$$

$$z_{k+1} = Cx_{k+1} + v_{k+1} \quad (2)$$

Where A, B, C are the system matrix, the input gain coefficient matrix and the measurement matrix. w_k is the system noise, v_k is the measurement noise.

KF is constructed as a predictor-corrector [28] with the assumption that the system noise and measurement noise are Gaussian-normally distributed and have zero mean. Equation (3) is shown in the state estimation prediction phase. Then the state error covariance is calculated according to Equation (4).

$$x_{k+1|k} = Ax_{k|k} + u_k \quad (3)$$

$$P_{k+1|k} = AP_{k|k}A^T + Q_k \quad (4)$$

In the update stage, the state error covariance is derived in Equation (4) to update the reformed covariance S_{k+1} and the Kalman gain K_{k+1} (Equations (5) and (6)). Equation (7) shows the Kalman gain coefficient used to update the estimated state. The state error covariance matrix is also updated using its previous value according to Equation (8).

$$S_{k+1} = C_{k+1}P_{k+1|k}C_{k+1}^T + R_{k+1}, \quad (5)$$

$$K_{k+1} = P_{k+1|k}C_{k+1}^T S_{k+1}^{-1}, \quad (6)$$

$$x_{k+1|k+1} = x_{k+1|k} + K_{k+1}\left(Z_{k+1} - C_{k+1}x_{k+1|k}\right), \quad (7)$$

$$P_{k+1|k+1} = \left(I - K_{k+1}C_{k+1}\right)P_{k+1|k}\left(I - K_{k+1}C_{k+1}\right)^T + K_{k+1}R_{k+1|k}K_{k+1}^T \quad (8)$$

where I refers to the identity matrix $n \times n$ (n is the number of states), k refer to the time step of the system, $k \mid k$ refers to the updated values, $k+1 \mid k$ refers to the predicted values at time $k+1$.

2.2. DeepSORT-based multi-object tracking

The DeepSORT algorithm is described in Figure 1 [29]. It is based on the method of comparing the motion characteristics of vehicles in consecutive frames, then calculating the similarity. At the same time, this algorithm also links vehicles in adjacent frames by combining information about their appearance characteristics and movement directions.

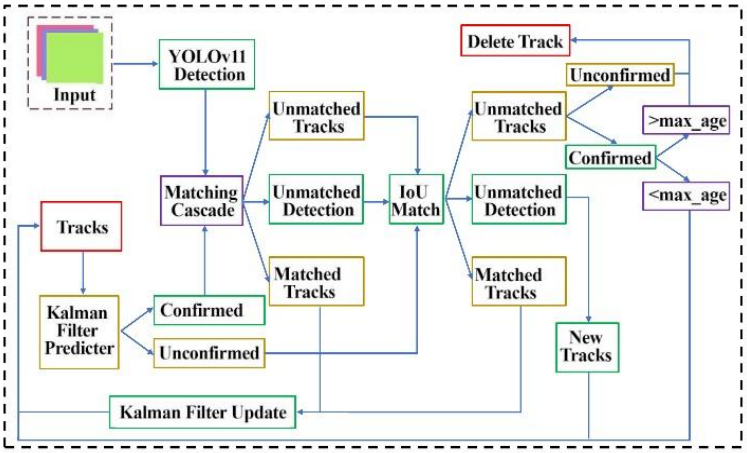


Figure 1. Vehicle tracking is based on DeepSORT algorithm

2.3. Slicing-aided inference for small object detection

As shown in Figure 2, dividing the input image into overlapping patches results in relatively larger pixel regions for small objects than the image fed into the network [30].

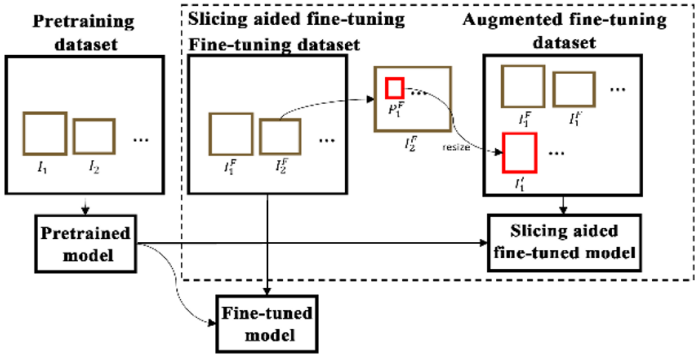


Figure 2. Illustration of the image slicing strategy used to enlarge small-object regions during inference

The image is segmented into smaller patches during inference, and predictions are derived from bigger, scaled versions of these patches (Figure 3).

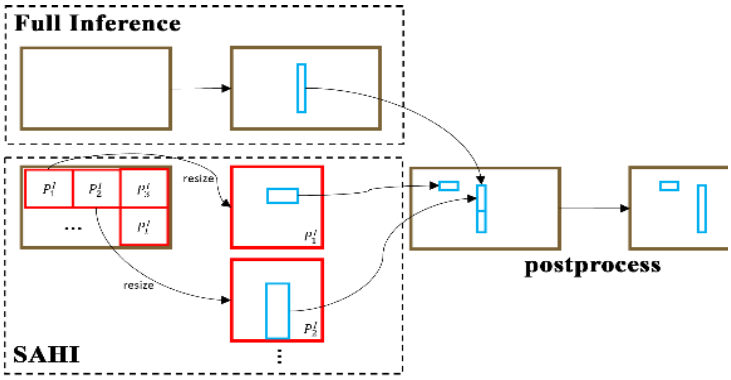


Figure 3. Workflow of the slicing-based inference process for patch-level object detection

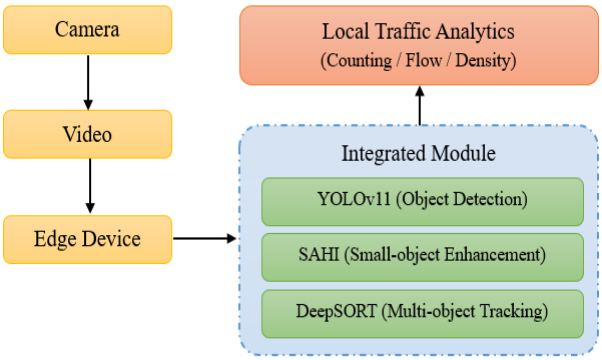
Per Image $I_1^F, I_2^F, ..., I_j^F$ cut into overlapping patch $P_1^F, P_2^F, ..., P_j^F$ with sizes M and N selected within a predefined range $[M_{\min}, M_{\max}]$ and $[N_{\min}, N_{\max}]$ are considered as hyperparameters. Then during the fine-tuning process, the image arrays are resized while maintaining a constant aspect ratio to obtain an improved image $I_1^{'}, I_2^{'}, ..., I_k^{'}$ where the object size is relatively larger than the original image. The images $I_1^{'}, I_2^{'}, ..., I_k^{'}$ along with original image $I_1^F, I_2^F, ..., I_j^F$ used during refinement. As the patch size decreases, larger objects may not fit into a single slice and the intersecting regions, and this may result in poor detection performance for larger objects.

The slicing method is also used in the inference step. First, the original query image I is sliced into l number of overlapping patches $M \times N$ $P_1^I, P_2^I, ..., P_l^I$. Each patch is then resized while the same aspect ratio is maintained. The object detection transition is applied independently to each overlapping patch. Optional full inference can be applied using the original image to detect larger objects. Finally, if the predicted results overlap, the FI results are merged back to the original size using Non-Maximum Suppression (NMS). In NMS, boxes with intersection-to-union (IoU) ratio higher than a predefined matching threshold T_m are matched.

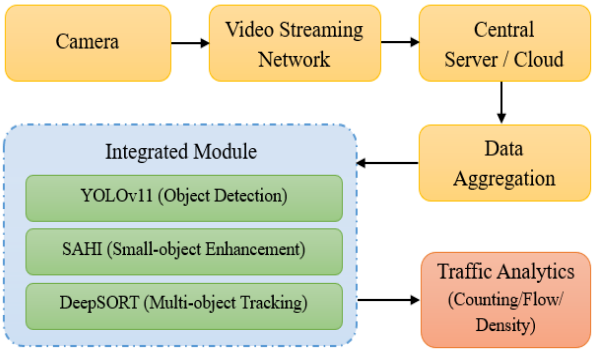
3. Proposed Multi-view Traffic Monitoring Framework using YOLOv11

3.1 System architecture

This section presents a proposed traffic monitoring framework to support urban traffic monitoring scenarios using multiple cameras in an urban environment. Video data is collected from multiple traffic cameras placed at different heights and viewing angles under real-world traffic conditions. To evaluate the feasibility of deployment under different infrastructure conditions, the study considers two processing architectures: edge-based processing and centralized processing, as illustrated in Figure 4.



(a) edge device-based processing architecture



(b) centralized processing architecture

Figure 4. Intelligent traffic monitoring system deployment architecture

In the edge-based architecture, video streams are processed locally at camera nodes or nearby edge devices. As illustrated in Figure 4(a), the detection and tracking process is executed directly on the edge hardware. This approach has the advantage of low response times and reduced network bandwidth usage, making it suitable for real-time applications. However, this solution requires replacing existing camera equipment or adding supporting edge devices, thus significantly increasing the overall cost of the system.

Conversely, a centralized architecture transmits video streams from multiple surveillance cameras to a central server or cloud platform for processing. As shown in Figure 4(b), the system performs data aggregation before applying the components in the proposed processing framework, including vehicle detection, multi-object tracking, and small object detection support. By leveraging high-performance computing resources, the centralized processing architecture is expected to deliver more stable processing performance and support traffic data management. However, this method requires large transmission bandwidth as well as high-performance server infrastructure.

Table 1. Comparison between edge device-based processing architecture and centralized processing architecture for intelligent traffic monitoring systems

Criteria	Edge Device-Based Processing	Centralized Processing
Processing location	At camera node or edge device	Central server or cloud
Latency	Low	Depends on network
Computational capability	Limited	High
Bandwidth	Low	High
System scalability	Limited	Highly scalable
Deployment cost	High	Reuse centralized infrastructure
Suitability	Suitable for local tasks	Highly suitable for large-scale traffic monitoring

Considering the characteristics of the existing infrastructure and the requirements of smart city traffic monitoring, a comparison is presented as in Table 1. Based on this, a centralized processing architecture was chosen for the proposed system framework as it aligns with the centralized management model commonly applied in current monitoring systems. This approach allows for the utilization of existing infrastructure and camera systems without requiring significant hardware changes.

3.2. YOLOv11 architecture for vehicle detection

YOLOv11 is a computer vision model architecture with several key improvements including improved feature extraction, higher accuracy with fewer parameters, and faster processing speed. YOLOv11 enables more accurate detection of small details even in difficult situations. The YOLOv11 model architecture is depicted in Figure 5.

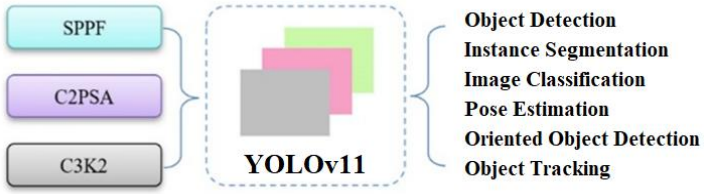


Figure 5. Network architecture of the YOLOv11 object detection model

YOLOv11 features significant improvements, including the C3K2 architecture replacing the C2F architecture used in previous versions to enhance computational efficiency, as well as a smaller kernel size contributing to faster processing while maintaining performance. The C2PSA block, added in YOLOv11, enhances spatial awareness in feature maps, thereby improving detection accuracy for objects of varying sizes and positions [31].

3.3. Vehicle Detection and multi-object tracking using YOLOv11 and DeepSORT

The combination of DeepSORT and YOLOv11 is used to build a function for detecting and tracking multiple objects in a traffic video sequence. In this configuration, YOLOv11 independently detects vehicles on each frame without maintaining object recognition between frames. DeepSORT is used to link the detection results over time by combining motion information and recognition features extracted by the deep learning network. The integrated model between YOLOv11 and DeepSORT is illustrated in Figure 6.

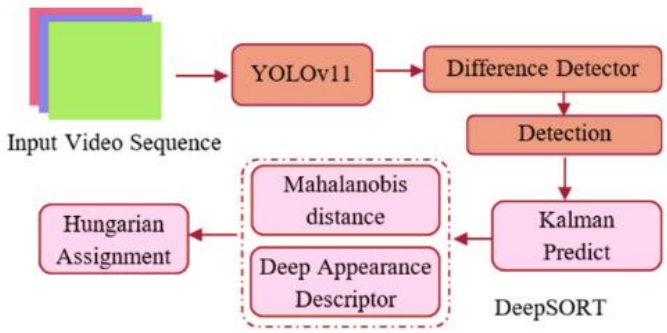


Figure 6. Integrated vehicle detection and tracking framework using YOLOv11 and DeepSORT

In the proposed system framework, YOLOv11 is used to detect and locate vehicles within each video frame. The detected objects are then passed to DeepSORT for multi-object tracking between consecutive frames. DeepSORT combines motion information and visual features to maintain the identity of each vehicle during movement. A KF is used to predict the object's position in the next frame, while a Hungarian algorithm performs the linking process between tracked objects and newly detected objects. Additionally, the IoU index is used to aid in assessing the similarity between bounding boxes during object assignment. Based on the linking results, the system updates existing tracks, creates new tracks for unassigned objects, and removes invalid tracks.

3.4. Enhancing Small-Object Detection Using YOLOv11 and SAHI

Vehicle detection at long distances remains a challenge for traffic monitoring systems, especially when data is collected from cameras mounted high up to extend the field of view. In these cases, vehicles often appear very small in the image. Therefore, in some situations, small or distant objects may be missed during the system's object detection process. To explore the possibility of overcoming this limitation, the study integrates SAHI techniques into the YOLOv11 inference process.

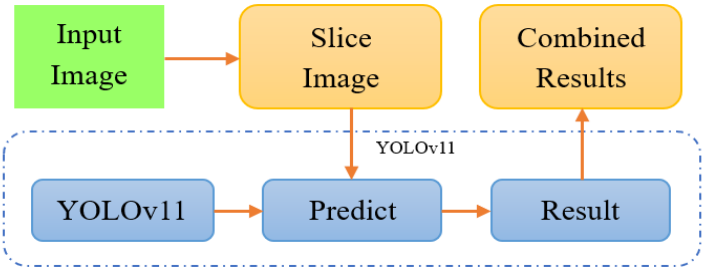


Figure 7. Slicing-aided inference process using YOLOv11 for small-object detection

This method allows the input image to be fragmented into multiple small, partially overlapping regions, and then object detection is performed on each region. The outputs are then combined to produce the final detection result. By focusing processing on smaller regions, this method is expected to aid in detecting small vehicles or vehicles at a distance in certain traffic situations. However, the increasing number of regions requires more computation, which can slow down inference speed and affect the overall performance of the system.

4. Experimental Results

4.1. Dataset and Evaluation Protocol

Experimental data were collected from urban traffic environments via a system of fixed surveillance cameras in Thai Nguyen City, Vietnam. As shown in Table 2, the dataset reflects varying observation conditions, including changes in distance, camera viewing angle, vehicle density, lighting conditions, and the presence of local obstacles, etc. The recorded objects mainly include motorcycles, cars, buses, trucks, bicycles, and pedestrians.

The dataset comprises numerous short video segments captured at frame rates common in traffic monitoring systems. The collected data was used for experiments related to vehicle detection, multi-object tracking, and vehicle counting in the surveyed traffic scenarios. A portion of the data was manually annotated for quantitative evaluation of detection and tracking tasks, while the remainder was used for qualitative evaluation of the system's performance.

In this study, the detector was initialized from pre-trained YOLOv11 weights and fine-tuned on annotated data to perform evaluation experiments. Simultaneously, experimental results were used to evaluate the detection accuracy, tracking sustainability, and processing performance of the system framework.

Table 2. Summary of Dataset Description

Component	Description
Collection Area	Thai Nguyen, Vietnam
Data Source	Surveillance cameras
Number of Cameras	~5
Video Duration	~1–3 minutes/segment
Frame Rate	~20–30 FPS
Object Categories	Motorcycle, Car, Bus, Truck, Bicycle, Pedestrian

The object detection results of the system based on different YOLO versions will affect efficiency and accuracy, thus affecting the overall performance of the system. This study uses YOLOv11 as the detector for experimental evaluation.

Frame performance is evaluated using metrics commonly applied to object detection and tracking, including mAP (Mean Average Precision) and FPS (Frames Per Second). P (Precision) is calculated using the formula $P = TP/(TP + FP)$, and R (Recall) is calculated using the formula $R = TP/(TP + FN)$. P reflects the accuracy of the predictions, while R represents the model's ability to detect all objects. The values TP (True Positive), FP (False Positive), and FN (False Negative) correspond to accurate identification, inaccurate identification, and

instances where the object exists but is not detected by the system, respectively.

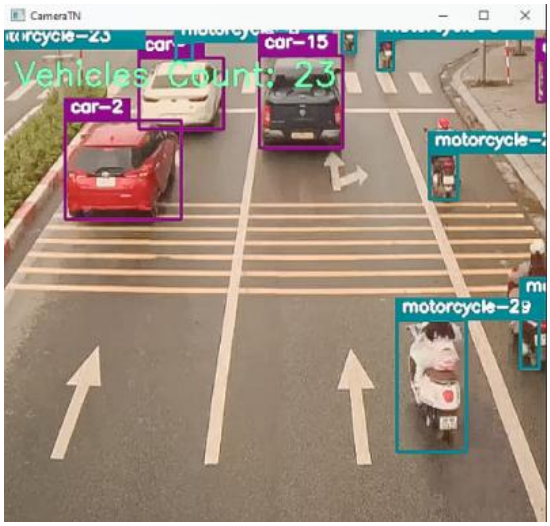
mAP is the primary metric used to evaluate the model's effectiveness in object and vehicle detection. This metric is calculated by averaging the AP (Average Precision) values across the entire Recall range, from 0 to 1. Detailed definitions of AP and mAP are presented in Equation 9.

$$AP = \int_0^1 P(r)dr \; ; \; mAP = \frac{1}{n}\sum_{k=1}^{n} AP_k \qquad (9)$$

In the following section, the test results of the system framework for traffic flow detection, monitoring, and analysis tasks are presented and discussed in detail.

4.2. Vehicle detection, tracking, and traffic flow analysis using YOLOv11 and DeepSORT

In the proposed integration framework, YOLOv11 is used for vehicle detection, while DeepSORT handles tracking in traffic video sequences. The system output includes time-stamped video and object identifiers, which are used for traffic flow analysis. During processing, the system assigns identifiers to detected objects and maintains this information between consecutive frames within the experimental conditions. An illustrative example is shown in Figure 8. Experiments were performed on an HP Z440 Workstation using an Intel Xeon E5-2696 v3 processor, 16 GB of VRAM graphics memory, 64 GB of RAM, and Windows 10 operating system.



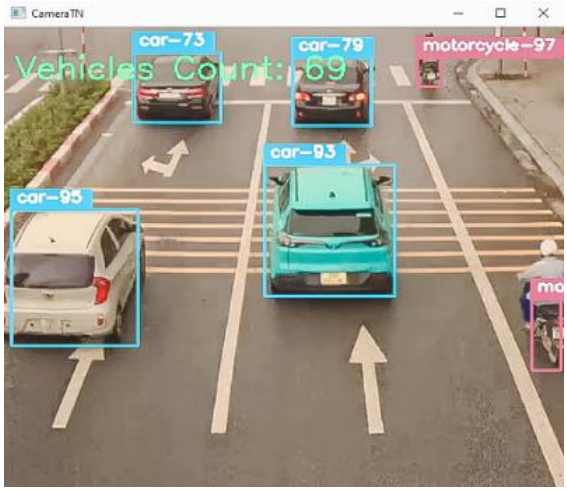


Figure 8. Examples of vehicle detection and tracking results using YOLOv11 and DeepSORT

In some cases, small vehicles may be partially or completely obscured by other objects (Figure 9), affecting the accuracy of detection and tracking. Combining YOLOv11 with DeepSORT and KF allows for linking objects between consecutive frames and estimating the object's state when observation data is interrupted for short periods. After processing is complete, the system outputs a video showing the detected and tracked objects, along with statistical information on the number of vehicles.

To ensure consistency of the experimental framework, the DeepSORT algorithm was kept constant across all survey configurations, while the detector was changed between YOLOv5 and YOLOv11 versions. Experimental results showed that the mAP@0.5 value increased from 0.58 for YOLOv5 to 0.66 for YOLOv11. This improvement may be due to YOLOv11's enhanced feature extraction capabilities, particularly in scenes containing vehicles of varying sizes and partially obscured. However, the improvement in detection quality was accompanied by a decrease in processing speed. Specifically, the processing speed decreased from 19.2 FPS for YOLOv5 to 10.1 FPS for YOLOv11. These results demonstrate the trade-off between detection accuracy and processing speed in the configurations examined.

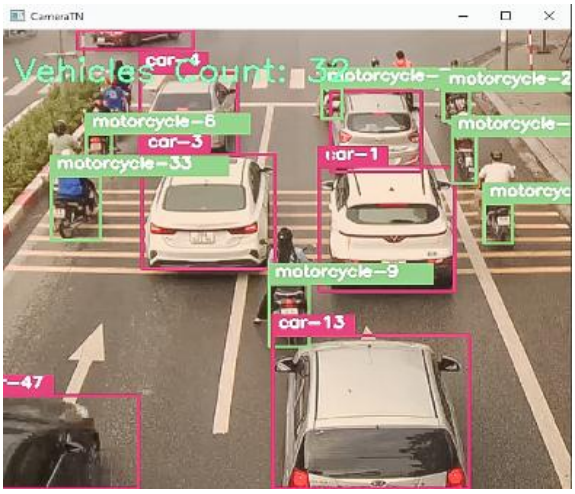


Figure 9. Tracking performance for small or partially occluded vehicles

This result is consistent with the trend reported in recent studies on YOLO-based vehicle detection [13, 14], in which newer versions of YOLO generally achieve higher accuracy than earlier versions. Although direct comparison is difficult due to differences in datasets and experimental conditions, the results obtained show the potential applicability of the proposed framework in line with the current development trend of DL-based traffic monitoring systems.

4.3. Enhancing small-object detection using YOLOv11 and SAHI

In urban traffic monitoring applications using high-positioned cameras, video data often contains many small objects such as pedestrians, motorcycles, and cars. As the observation distance increases, the size of objects in the image decreases significantly, increasing the difficulty of the object detection problem. To further assess this challenge, the study conducted an experimental configuration using the YOLOv11 model combined with the SAHI technique to evaluate its ability to support the detection of small objects in traffic data. Experiments were performed on the Google Colab platform with a T4 GPU and 15 GB of RAM. Experimental parameters are presented in Table 3.

Table 3. Details of experimental parameters set

Components	Description
Model type YOLO uses	YOLOv11
Slice size	256 × 256 Pixels
Overlap ratio	25%
Confidence threshold	0.25

Figure 10. Visually display the results of small object detection using YOLOv11 and SAHI

Experimental results show that the YOLOv11 configuration combined with SAHI recorded a higher number of detected objects in several observation scenarios, while reducing the omission of objects appearing at a distance or small in size in the image (Figure 10). Observations from experimental examples show that vehicles appearing at relatively long distances can still be detected with confidence levels ranging from 0.41–0.71, while some vehicles at close distances recorded confidence levels up to 0.87. Additionally, some small objects such as motorcycles and pedestrians were also detected with confidence values ranging from 0.34–0.62. Quantitative evaluation results show that integrating SAHI improves detection performance compared to the configuration using only YOLOv11, with mAP@0.5 increasing from 0.66 to 0.70 and AP_S increasing from 0.29 to 0.40. The increase in observability aligns with SAHI's design objective, which is to expand the efficient representation of small objects through inference based on component image regions. These results suggest that combining SAHI with YOLOv11 can help improve the ability to observe small objects within the range of experimental conditions investigated. Furthermore, these results are consistent with previous studies on SAHI techniques [20, 21], and also show that image splitting techniques can still support the detection of small vehicles in traffic data collected from wide-angle surveillance cameras.

4.4. Discussion of experimental results

Experimental results show that the YOLOv11 framework combined with DeepSORT can be used to detect, track, and statistically analyze vehicles in traffic video sequences. The system's output includes object detection results, identification information maintained between consecutive frames, and the number of vehicles recorded.

During tracking, KF is used to estimate the object's state between consecutive frames. This mechanism helps

maintain tracking in some cases where the object is obscured or only partially visible in the frame. For objects that temporarily leave the observation area for a short time, tracking information is still maintained. Therefore, it can help limit the phenomenon of repeated counting when the object reappears. The "Number of Vehicles" parameter represents the total number of objects recorded in the observation area throughout the video processing, including objects that are no longer visible at the current time. This information can be used as a reference indicator during traffic density and flow monitoring. However, the system's performance still depends on the quality of the input data, including image resolution and camera recording conditions. This observation is also consistent with previous studies on YOLO-based vehicle detection and tracking [18, 19], in which the combination of detection models and tracking algorithms helps maintain object recognition between successive frames and supports traffic analysis tasks.

For the YOLOv11 configuration combined with SAHI, an increase in the number of detected objects was observed in some scenarios, especially for small objects or those appearing at a distance. At the same time, the number of missed objects observed was lower compared to the YOLOv11-only configuration in some experimental cases. The results also show that detection performance is affected by many factors, including camera placement, viewing angle, lighting conditions, and environmental changes. The results obtained are generally consistent with previous studies on SAHI [20, 21]. In the context of this study, the observed improvement in AP_S suggests that image separation techniques can help enhance the detection of small vehicles in traffic data collected from surveillance cameras with wide viewing angles and long viewing distances.

5. Conclusions and Future Work

This study presents an integrated framework for urban traffic monitoring, combining the YOLOv11 object detection model, the DeepSORT multi-object tracking algorithm, and the SAHI technique. In experimental scenarios, the system framework was evaluated under common urban traffic data situations, including local occlusion, object size changes, and traffic density fluctuations. Under established experimental conditions, the YOLOv11 configuration combined with DeepSORT achieved a processing speed of approximately 10.1 FPS; for object detection tasks, the model achieved an mAP@0.5 of 0.66. For the integrated SAHI configuration, observed results show that the number of detected objects tends to increase in some scenarios containing small objects or objects appearing at a distance, with an AP_S index of 0.40. Overall, the experimental results obtained are consistent with trends reported in recent studies on YOLO-based vehicle detection and tracking, as well as techniques supporting small object detection. Within the scope of the survey conditions, the proposed framework shows potential to support vehicle detection, tracking, and statistical tasks on real-world traffic data.

The research framework is built on the basis of exploiting data from existing surveillance camera systems, thereby allowing consideration of the applicability of object detection and tracking techniques in urban traffic monitoring. However, the current assessment is limited to a relatively small number of monitoring locations and traffic conditions. In the future, expanding the data collection scope to include more traffic areas and observation conditions could contribute to a more comprehensive evaluation of the system framework in diverse contexts. Furthermore, optimizing the processing configuration to balance inference speed and detection capability remains a worthwhile area of research. Beyond vehicle detection and tracking tasks, the current research framework could also be considered for expansion to higher-level traffic analysis problems, such as flow analysis, vehicle density statistics, or tracking traffic flow variability over time.

Acknowledgments

The authors would like to thank Thai Nguyen University of Technology (TNUT), Viet Nam for the support.

References

[1] Rathore SP, Singh J, Bhatt PN, Kumar BV, Haldorai A, Muthusundari S. Intelligent Congestion Management Through AI-Based Traffic Monitoring and Dynamic Signal Timing. In2025 Modern Electronics Devices and Intelligent Communication Systems (MEDCOM) 2025 Dec 11 (pp. 1008-1012). IEEE.

[2] Mien TL, Vu DV. Development of the System of Monitoring Traffic Vehicle Volume and Density on Vietnam's Streets. In: 2022 11th International Conference on Control, Automation and Information Sciences (ICCAIS); 2022 Nov 21–24; pp. 292–297. IEEE.

[3] Khan H, Thakur JS. Smart traffic control: machine learning for dynamic road traffic management in urban environments. Multimedia Tools and Applications. 2025 Apr;84(12):10321-45.

[4] Tahir M, Qiao Y, Kanwal N, Lee B, Asghar MN. Real-time event-driven road traffic monitoring system using CCTV video analytics. IEEE Access. 2023 Dec 7;11:139097-111.

[5] Musa AA, Malami SI, Alanazi F, Ounaies W, Alshammari M, Haruna SI. Sustainable traffic management for smart cities using internet-of-things-oriented intelligent transportation systems (ITS): challenges and recommendations. Sustainability. 2023 Jun 21;15(13):9859.

[6] Krishna, Mohan A., P. V. N. Reddy, and K. Satyma Prasad. "Effective Object detection and Tracking using Attention-Driven YOLO v9 Model with Multi-Stage Cascaded Convolutional Model." EAI Endorsed Transactions on Internet of Things 11 (2024).

[7] Hua HT, Nguyen NT, Dinh NV, Nguyen HT, Nguyen MT. A Comprehensive Survey of AI-empowered Multiple Robot Systems: Development and Research Challenges. ICSES Trans. Comput. Netw. Commun.. 2021.

[8] Nguyen MT, Truong LH, Le TT. Video surveillance processing algorithms utilizing artificial intelligent (AI) for

unmanned autonomous vehicles (UAVs). MethodsX. 2021 Jan 1;8:101472.
[9] Duong-Trung, Hieu, and Nghia Duong-Trung. "Integrating YOLOv8-agri and DeepSORT for advanced motion detection in agriculture and fisheries." EAI Endorsed Transactions on Industrial Networks and Intelligent Systems 11, no. 1 (2024): e4.
[10] Mien TL, Nguyen TD, Nguyen LV. Deploying YOLOv8 for Real-Time Road Crack Detection on Smart Road Length Measurement Devices. Journal of Future Artificial Intelligence and Technologies. 2025 May 29;2(1):135–144.
[11] Jiménez-Bravo DM, Murciego ÁL, Mendes AS, San Blás HS, Bajo J. Multi-object tracking in traffic environments: A systematic literature review. Neurocomputing. 2022 Jul 14;494:43-55.
[12] Nguyen MD, Nguyen MT. Artificial intelligence for human detection, identification and tracking: Methods and applications. Journal of Future Artificial Intelligence and Technologies. 2025 Apr 30;2(1):79-94.
[13] Nayyer, Aadarsh, Abhinav Kumar, Aayush Rajput, Shruti Patil, Pooja Kamat, Shivali Wagle, and Tanupriya Choudhury. "Color-driven object recognition: a novel approach combining color detection and machine learning techniques." EAI Endorsed Transactions on Internet of Things 10 (2023).
[14] Krishna, Mohan A., P. V. N. Reddy, and K. Satyma Prasad. "Effective Object detection and Tracking using Attention-Driven YOLO v9 Model with Multi-Stage Cascaded Convolutional Model." EAI Endorsed Transactions on Internet of Things 11 (2024).
[15] Do HT, Truong LH, Nguyen MT, Chien CF, Tran HT, Hua HT, Nguyen CV, Nguyen HT, Nguyen NT. Energy-Efficient Unmanned Aerial Vehicle (UAV) Surveillance Utilizing Artificial Intelligence (AI). Wireless Communications and Mobile Computing. 2021;2021(1):8615367.
[16] Hassaballah M, Kenk MA, Muhammad K, Minaee S. Vehicle detection and tracking in adverse weather using a deep learning framework. IEEE transactions on intelligent transportation systems. 2020 Sep 2;22(7):4230-42.
[17] Shobha BS, Deepu R. Deep learning assisted active net segmentation of vehicles for smart traffic management. Global Transitions Proceedings. 2021 Nov 1;2(2):282-6.
[18] Nataraj, Chandrasekharan, Lawrence Wong Ming Wei, Mukil Alagirisamy, and Sathish Kumar Selvaperumal. "Design Of Intelligent Road Eye Using AI And Machine Learning For Automobiles." EAI Endorsed Transactions on Internet of Things 11 (2024).
[19] Lin Q, Zhang S, Xu S. Construction of Traffic Moving Object Detection System Based on Improved YOLOv5 Algorithm. In2023 2nd International Conference on 3D Immersion, Interaction and Multi-sensory Experiences (ICDIIME) 2023 Jun 27 (pp. 268-272). IEEE.
[20] Guan X, Guan Z, Zhu S, Chen B. Research on the application of YOLOv8 model based on ODConv and SAHI optimization in dense small target crowd detection. In2024 IEEE 2nd International Conference on Control, Electronics and Computer Technology (ICCECT) 2024 Apr 26 (pp. 726-732). IEEE.
[21] Vu TC, Nguyen TV, Nguyen TV, Nguyen DT, Dinh LQ, Nguyen MD, Nguyen HT, Nguyen HT, Nguyen MT. Object detection in remote sensing images using deep learning: From theory to applications in intelligent transportation systems. Journal of Future Artificial Intelligence and Technologies. 2025 Jun 24;2(2):227-41.
[22] Ji H, Gao Z, Mei T, Ramesh B. Vehicle detection in remote sensing images leveraging on simultaneous super-resolution. IEEE Geoscience and Remote Sensing Letters. 2019 Aug 8;17(4):676-80.
[23] Wang J, Wang Y, Wu Y, Zhang K, Wang Q. FRPNet: A feature-reflowing pyramid network for object detection of remote sensing images. IEEE Geoscience and Remote Sensing Letters. 2020 Dec 8;19:1-5.
[24] Zhang S, Mu X, Kou G, Zhao J. Object detection based on efficient multiscale auto-inference in remote sensing images. IEEE Geoscience and Remote Sensing Letters. 2020 Jul 3;18(9):1650-4.
[25] Yang X, Sun H, Sun X, Yan M, Guo Z, Fu K. Position detection and direction prediction for arbitrary-oriented ships via multitask rotation region convolutional neural network. IEEE access. 2018 Sep 13;6:50839-49.
[26] Wang P, Sun X, Diao W, Fu K. FMSSD: Feature-merged single-shot detection for multiscale objects in large-scale remote sensing imagery. IEEE Transactions on Geoscience and Remote Sensing. 2019 Dec 11;58(5):3377-90.
[27] Sorenson HW. Least-squares estimation: from Gauss to Kalman. IEEE spectrum. 2009 Aug 25;7(7):63-8.
[28] Avzayesh M, Abdel-Hafez M, AlShabi M, Gadsden SA. The smooth variable structure filter: A comprehensive review. Digital Signal Processing. 2021 Mar 1;110:102912.
[29] Pujara A, Bhamare M. DeepSORT: Real time & multi-object detection and tracking with yolo and tensorflow. In2022 International conference on augmented intelligence and sustainable systems (ICAISS) 2022 Nov 24 (pp. 456-460). IEEE.
[30] Akyon FC, Altinuc SO, Temizel A. Slicing aided hyper inference and fine-tuning for small object detection. In2022 IEEE international conference on image processing (ICIP) 2022 Oct 16 (pp. 966-970). IEEE.
[31] Kang S, Hu Z, Liu L, Zhang K, Cao Z. Object detection YOLO algorithms and their industrial applications: Overview and comparative analysis. Electronics. 2025 Mar 11;14(6):1104.