Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.067650

ARTICLE

BES-Net: A Complex Road Vehicle Detection Algorithm Based on Multi-Head Self-Attention Mechanism

Heng Wang[1] **and Jian-Hua Qin**[2,*]

[1]Key Laboratory of Advanced Manufacturing and Automation Technology, Guilin University of Technology, Education Department of Guangxi Zhuang Autonomous Region, Guilin, 541006, China
[2]College of Mechanical and Control Engineering, Guilin University of Technology, Guilin, 541004, China

*Corresponding Author: Jian-Hua Qin. Email: qinjh2@sina.com

Received: 08 May 2025; Accepted: 24 June 2025; Published: 29 August 2025

ABSTRACT: Vehicle detection is a crucial aspect of intelligent transportation systems (ITS) and autonomous driving technologies. The complexity and diversity of real-world road environments, coupled with traffic congestion, pose significant challenges to the accuracy and real-time performance of vehicle detection models. To address these challenges, this paper introduces a fast and accurate vehicle detection algorithm named BES-Net. Firstly, the BoTNet module is integrated into the backbone network to bolster the model's long-distance dependency, address the complexities and diversity of road environments, and accelerate the detection speed of the BES-Net network. Secondly, to accommodate the varying sizes of target vehicles, the efficient multi-scale attention mechanism (EMA) was added to enrich feature information and enhance the model's expressive power by combining features from multiple scales. Finally, the Slide loss function is specifically designed to give higher weight to difficult samples, thereby improving the detection accuracy of small targets. The experimental results demonstrate that BES-Net achieves superior performance compared to conventional object detection models, with mAP50 (mean Average Precision), precision, and recall reaching 73.2%, 74.8%, and 73.1%, respectively. These metrics represent significant improvements of 8.5%, 7.2%, and 11.7% over the baseline network. This significant improvement underscores the high robustness of the BES-Net model in vehicle detection tasks. In addition, the BES-Net network has been deployed on NVIDIA Jetson Orin NX equipment, providing a solid foundation for its integration into intelligent transportation systems. This deployment not only showcases the practicality of the model but also highlights its potential for real-world applications in autonomous driving and ITS.

KEYWORDS: Vehicle detection; YOLOv8; MHSA; EMA

1 Introduction

Vehicle detection is a subclass of object detection, utilizing computer vision technology to accurately identify the type and location of vehicles in images or videos [1]. It serves as the foundation for subsequent tasks such as vehicle tracking and traffic statistics, and is a core component of intelligent transportation systems. Efficient and accurate vehicle detection technology is crucial for improving traffic safety and optimizing traffic management [2]. Currently, applying object detection algorithms to traffic scenarios poses two major challenges.

Traditional vehicle detection algorithms can provide more effective solutions in simple scenarios, especially in scenarios with limited computing resources and high real-time requirements, and these methods still have certain application values [3]. However, with the rapid development of deep learning methods, vehicle detection methods based on convolutional neural networks (CNN) have gradually replaced

the traditional algorithms. Especially in complex scenarios, deep learning can automatically learn features and significantly improve detection accuracy and robustness [4].

The model designed in this paper incorporates targeted improvements to address the challenges of real-world road vehicle detection in complex environments. First, ensuring the accuracy of the detection algorithm is crucial. The traffic scenes are fraught with variable environmental factors like lighting changes and weather conditions, which impede detection precision and necessitate enhanced algorithmic robustness [5]. To address this, we introduced Slide Loss into our model. By assigning higher weights to hard samples, it improves detection accuracy in complex environments. To ensure the algorithm maintains high stability and reliability across varying vehicle sizes [6], we integrated EMA into the model's backbone network. This enhances multi-scale feature fusion capabilities, enabling the model to focus more effectively on vehicle characteristics and thereby boost detection accuracy.

The design of the model framework must balance detection accuracy with algorithmic complexity to ensure efficient operation on edge devices with limited memory and computational resources, achieving both real-time and precise vehicle detection. Consequently, developing and applying high-precision, fast-detection vehicle technology has become an urgent need. To tackle this, we incorporated BoTNet, a module combining self-attention mechanisms with CNNs, at the bottom of the model's backbone network. This enhances the network's feature extraction capability and accelerates training convergence. Furthermore, deploying the model on NVIDIA Jetson Orin NX devices demonstrates the integration of deep learning with embedded hardware, laying a foundation for establishing future intelligent transportation platforms.

The rest of this paper is organized as follows: Section 2 provides an overview of the literature, analyzing the current state of deep learning-based vehicle detection. Section 3 introduces the overall architecture of our model and the improved modules. Section 4 presents the experimental results along with corresponding analyses. Sections 5 and 6 contain the discussion and conclusion, respectively, outlining the research findings and potential directions for future studies.

2 Literature Review

The continuous development of the automotive industry has led to increasingly severe road traffic congestion, making the construction of intelligent traffic management systems critically important. Methods and technologies based on machine learning (ML) and deep learning (DL) can enhance vehicle safety and reduce congestion. Concurrently with technological progress, electric vehicles (EVs) have become a dominant presence on roads, necessitating effective management strategies for them. Mazhar et al. [7,8] employed various deep learning approaches, such as long short-term memory (LSTM) and CNN, to conduct tailored experiments for different EV types. Their work guides EVs towards charging stations and uploads relevant information to cloud platforms. Similarly, Ditta et al. utilized deep learning techniques for license plate recognition, transmitting vehicle detection data via the Internet of Things (IoT) [9].

While these methods contribute to alleviating road congestion, real-time detection of vehicles in complex road environments is essential to further control traffic flow and effectively reduce congestion.

At present, vehicle detection algorithms based on deep learning can be divided into two-stage detection algorithms and single-stage detection algorithms according to the process [10]. The two-stage algorithm first generates candidate regions and then performs classification and boundary box regression on these candidate regions [11]. As a result, the implementation process of the algorithm is complicated and the detection speed is slow, which brings difficulties to real-time vehicle detection. The single-stage target detection algorithm uses the end-to-end convolutional neural network to directly input the category and location information of the predicted target in the image, which improves the computational efficiency and real-time performance [12].

Therefore, the single-stage algorithm has great advantages in many practical applications, especially in scenarios with high real-time requirements [13]. In recent years, in order to meet the development needs of intelligent transportation systems, the object detection algorithm has been continuously improved in the traffic object detection task.

Although the performance of the new object detection algorithm in large public data sets is getting better and better, in the real complex traffic scene, factors such as vehicle occlusion, different sizes, and lighting changes may lead to serious problems such as vehicle missed detection and false detection. To address these issues, the researchers took into account the vehicle features extracted by the enhancement network. For example, Song et al. [14] employed MixUp and Mosaic data augmentation techniques to enhance the network's ability to learn local vehicle features and introduced efficient channel attention (ECA) mechanisms in the backbone network to better focus on key features. Eventually, they used a complex decoupled header to make predictions. Bochkovskiy et al. [15] not only use CSPDarknet53 as the backbone network to reduce network computation and memory consumption, but also utilize spatial pooling pyramid (SPP) and path aggregation network (PAN) to enhance feature fusion capability. On this basis, Jocher [16] proposed that YOLOv5 achieves faster inference speed by optimizing label allocation strategy and improving SPP module. Li et al. [17] and Xu et al. [18] used the heavily parameterized module RepVGG [19] block to improve the feature representation capability of small networks with similar inference speeds.

3 BES-Net Network Structure

YOLOv8n is the lightest version in the YOLOv8 series, with fewer parameters and lower computational requirements, enabling efficient operation on resource-constrained devices. Compared to YOLOv5 and YOLOv7, it is more suitable for real-time processing and resource-limited application scenarios, offering faster speeds, lower latency, and reduced computational resource consumption. In this paper, the model based on YOLOv8n proposes the following targeted improvements. Considering the complex road environment of vehicles in the dataset, in order to enhance the detection ability of the model for vehicles in the complex environment, the BoTNet module is added to the network backbone to enhance the long-distance dependence of the model and also increase the detection speed of the network [20]. In addition, we integrate EMA into the Neck part of the network to enhance the feature fusion capability of the network. Finally, aiming at the problem of small targets and occlusion at a long distance, the Slideloss function is introduced to solve the imbalance between easy and difficult samples. Fig. 1 shows the specific structure of the BES-Net network model. Through these improvements, the BES-Net model can detect target vehicles more effectively and meet the requirements of high-precision.

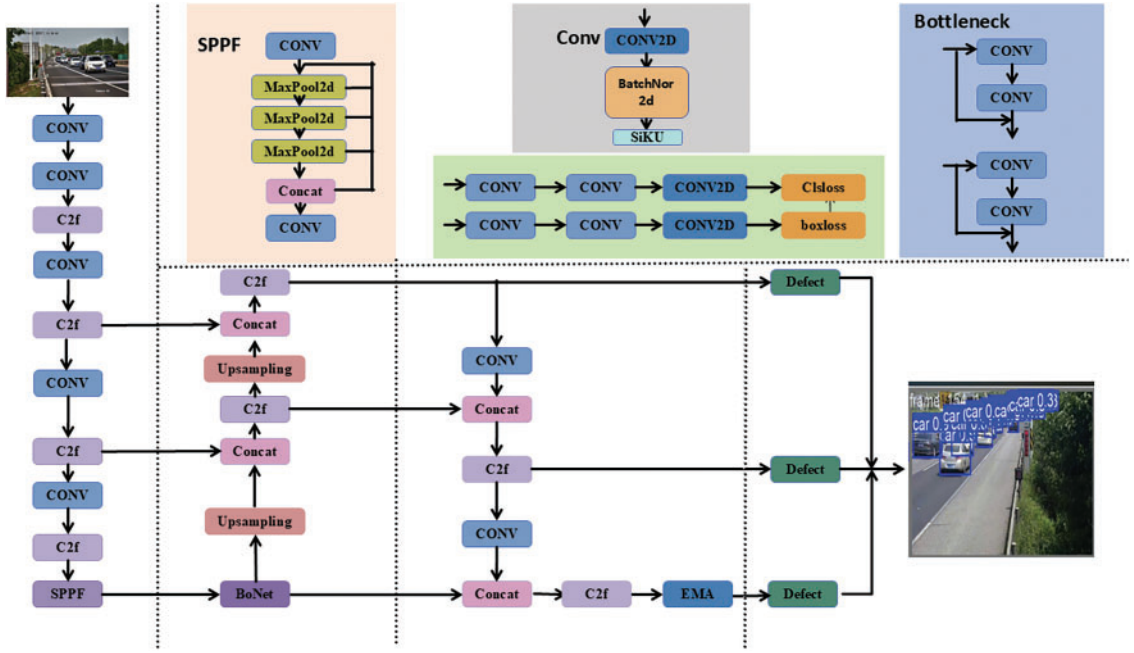

Figure 1: BES-Net network structure diagram

3.1 Pseudo Code of the Study

The pseudo-code of this study is outlined in Algorithm 1 below.

Algorithm 1: Pseudocode of the study

Step1: Preprocess the data and input it
Step2: Load the smoothed weights saved during the training phase through EMA
Step3: focus on hard samples within the Slide loss,
Step4: Generate classification probabilities and regression offsets through the YOLOv8n backbone network
Step5: Coordinate decoding: Convert the offsets to the original image coordinates
Step6: Retain the confidence above the threshold
Step7: Remove highly overlapping redundant boxes
Step8: Output the final results and overlay the detection boxes and class labels on the original image
Step9: Deploy the algorithm on edge devices for complex road vehicle detection

3.2 BoTNet Multi-Head Self-Attention Network

To address challenges such as complex road environments and uneven vehicle distribution, we introduce a BoTNet module—a conceptually simple yet powerful backbone network that integrates self-attention mechanisms. Built upon the ResNet architecture, BoTNet replaces the 3 × 3 convolutional layers in the last three bottleneck blocks with multi-head self-attention (MHSA) layers, a design choice that significantly enhances performance in object detection tasks. BoTNet employs relative position encoding, enabling the self-attention mechanism to perceive spatial relationships. This encoding approach not only considers content information but also effectively correlates features across different positions.

As illustrated in Fig. 2, the core of BoTNet is the MHSA mechanism. MHSA, an extension of the attention mechanism widely adopted in Transformer models, operates multiple independent attention mechanisms in parallel to capture attention distributions across different subspaces of the input sequence. This design allows for a more comprehensive understanding of potential semantic relationships within the sequence. By leveraging this mechanism, BoTNet achieves superior feature representation capabilities, particularly beneficial for detecting vehicles in complex and variable road environments.

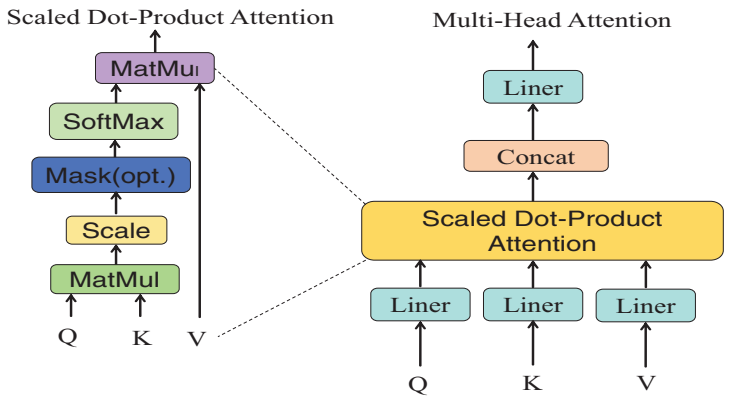

Figure 2: General schematic diagram of MHSA

In multi-head attention, the input sequence is first passed through three different linear transformation layers to obtain the Query, Key, and Value. Then, these transformed vectors are divided into several "heads", each with its own independent Query, Key, and Value matrices. For each head, a Scaled Dot-Product Attention operation is performed.

For each input feature $x_i \in \mathrm{X}$, we calculate the corresponding query, key and value through the weight matrix W_Q, W_K, W_V, which is computed as shown in Eqs. (1)–(3).

$$Q = XW_Q, \tag{1}$$

$$K = XW_K, \tag{2}$$

$$V = XW_V, \tag{3}$$

where W_Q, W_K and W_V are the learnable parameter.

The core of the self-attention mechanism is to calculate the correlation between the query and the key, usually through the dot product. Q For the query and key K, we first compute their dot product and then normalize it via the *softmax* function as shown in Eq. (4).

$$Attention\,(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right), \tag{4}$$

where $\frac{1}{\sqrt{d_k}}$ is a scaling factor used to avoid situations where the dot product value is too large to cause the gradient to disappear or the gradient to explode.

3.3 EMA Attention Mechanism

In order to solve the problem of poor network generalization caused by obvious differences between individual vehicles in self-built data sets, this paper introduces EMA based on cross-space learning. This module focuses on retaining information of each channel and reducing computing overhead, readjusting some channels into batch dimensions, and grouping channel dimensions into multiple sub-features. So that spatial semantic features are evenly distributed within each feature group. Thus, dependencies between features are captured more comprehensively [21]. The EMA attention mechanism is shown in the Fig. 3 below.

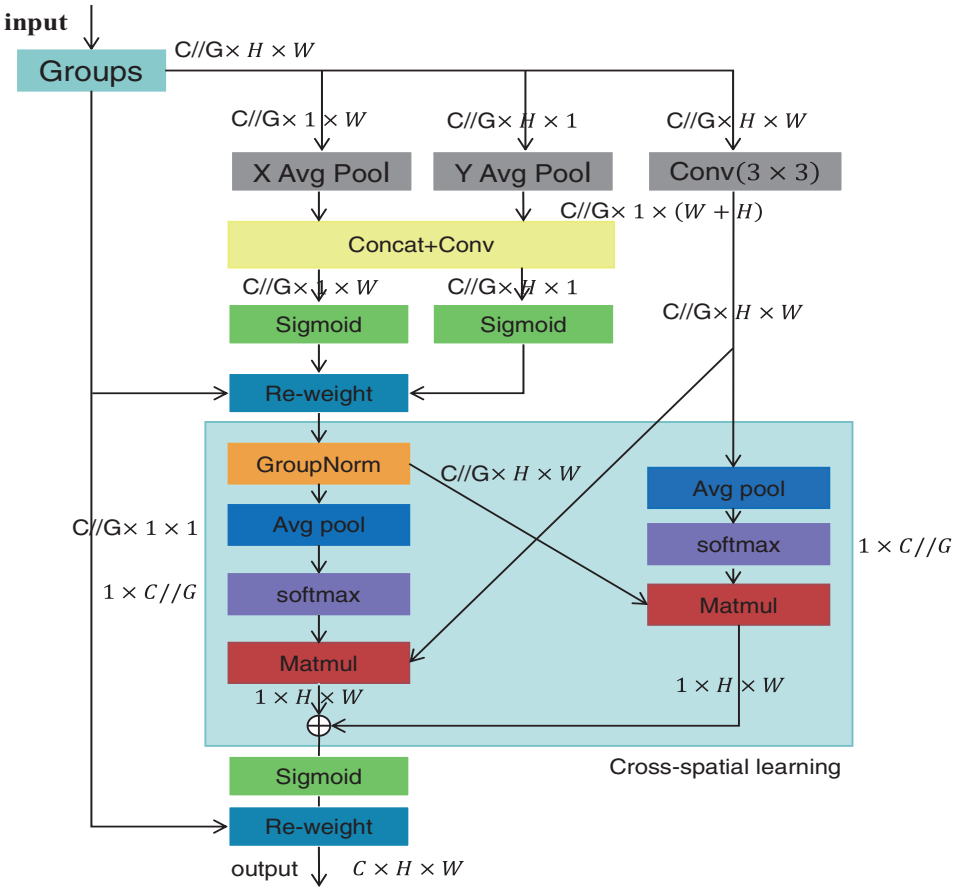

Figure 3: Structure of EMA attention mechanism

The EMA mechanism can learn effective channel descriptions without reducing the channel dimensionality during convolution operations and generate better pixel-level attention for high-level feature maps. Specifically, it selects the shared components of the 1 × 1 convolution from the CA module, which is named the 1 × 1 branch in EMA. To aggregate multi-scale spatial structural information, a 3 × 3 kernel is placed in parallel with the 1 × 1 branch to achieve rapid response, and this is referred to as the 3 × 3 branch. Considering feature grouping and multi-scale structures, effectively establishing short-term and long-term dependencies is beneficial for achieving better performance.

3.4 Slide Loss Function

Under the same camera perspective of vehicles in complex road environment in self-built data set, vehicles will be blocked by different vehicles, increasing the difficulty of identification. Therefore, the Slide Loss function is introduced in this study, which aims to solve the imbalance between easy samples and difficult samples. Easy samples and difficult samples are distinguished according to the IoU size between the prediction box and the real box, and the Slide weighting function is used to assign higher weights to difficult samples. Here is the Slide Loss formula were shown in Eq. (5).

$$L_{SlideLoss} = -\sum_{i,j} \{\omega_{i,j} \left(y_{i,j} \log\left(p_{i,j}\right) + \left(1 - y_{i,j}\right) \log\left(1 - p_{i,j}\right)\right), \tag{5}$$

where $L_{SlideLoss}$ indicates total loss. i, j is the index of each pixel in the image. $y_{i,j}$ is the true label of the (i, j) pixels in the image, with values of 0 or 1, representing the background (0) or the target area (1). $p_{i,j}$ is the prediction probability of the model for the target region of the (i, j) pixel, usually made with a CNN. i, j is the weight of the (i, j) pixel, set according to whether it is the target region or the background region.

4 Experimental Results and Analysis

4.1 Picture Annotation

This paper uses Labelme software to label vehicle types. Labelme is a widely used image labeling tool. The tool allows users to draw bounding boxes, polygons, and other shapes for objects in an image through a simple graphical interface and add semantic labels to them [22]. In this study, vehicles on the highway are labeled in 3 common categories, namely, cars, buses and trucks. Fig. 4 shows the data set labeling.

Figure 4: Photos of part of the data set

4.2 Data Set Production

The data set constructed in this paper, as shown in Fig. 4, consists of 3345 images, and the data set is divided into 2676 training images and 578 verification images. The photos in this data set are frame extracted from the surveillance video on the Nanjing expressway in China. The image resolution in all datasets has been standardized to 1920 × 1080 pixels to ensure high-quality image details. Some images have been resized to 256 × 256 pixels to meet the model training requirements. Additionally, we have performed a three-class classification on the vehicles in the images, categorizing them as cars, trucks, and buses. We have selected the Hikvision DS-2DE4220IW-DE for 360-degree panoramic monitoring, which features high resolution.

The selected time is during holidays. The characteristics of this data set are that traffic roads are crowded, the environment is complex, vehicle occlusion is serious, and multiple cameras of different angles and sizes of vehicles are covered, which is difficult to detect. It is suitable for vehicle detection data set in complex road environment. In order to increase feature diversity and prevent overfitting, a series of data enhancement methods, including adjusting image saturation, mirroring, etc. were employed in this study to expand the sample size and enhance the generalization ability and robustness of the model.

4.3 Blur and Noise Removal in Images

To validate the model's generalization capability, robustness, and accuracy in real-world applications, the following tests were conducted: normal images were blurred with a radius of 5–10 pixels, motion blur was set to 10–20 pixels, jitter noise intensity was adjusted to 5%–10%, and a slight displacement of 1–5 pixels was applied.

4.4 Evaluation Index of Vehicle Detection

We evaluated the size of the model by the number of parameters and the amount of calculation. The number of arguments represents the spatial complexity of the model, while the computation refers to the number of floating-point operations performed per second. The inference speed of the model is measured by FPS (frames per second). When calculating reasoning speed for small size models, we make sure to maximize the performance of hardware devices such as graphics cards, ultimately calculating FPS by the inspection time per batch. This method allows for a fairer and more accurate comparison of reasoning speeds for models of different sizes. To evaluate model performance, we use Precision, Recall, mAP50 and MAP50-95 as the main indicators. Precision and Recall are used as the basic indicators, and the final evaluation is determined by calculating the Average Precision (AP) to measure the recognition accuracy of the model. The calculation formula is shown in Eqs. (6)–(8).

$$Precision = \frac{TP}{TP + FP} \times 100\%, \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \times 100\%, \tag{7}$$

$$mAP = \frac{\sum_{i=1}^{k} \{APi}{k} \times 100\%. \tag{8}$$

4.5 Experimental Configuration

The open source PyTorch deep learning framework is used in this paper. The CPU uses the 14th generation Intel Core i7-14650H, the main frequency is 4.70 GHz; Window11 operating system, which includes Python 3.8 and CUDA 12.0; The graphics processor uses GeForce GTX 4060 with 12 GB of video memory. In order to adapt to our own data set, this paper retrained YOLOv8, optimized the training parameters and Batch Size, and did not use any pre-trained model in the training process. The input image size was set to 640 × 640 pixels, the model was trained 100 times, and the batch size was 16. The initial learning rate is 0.01. Random gradient descent (SGD) was used to optimize the parameters. The results from training and evaluating the BES-Net model are displayed in Fig. 5.

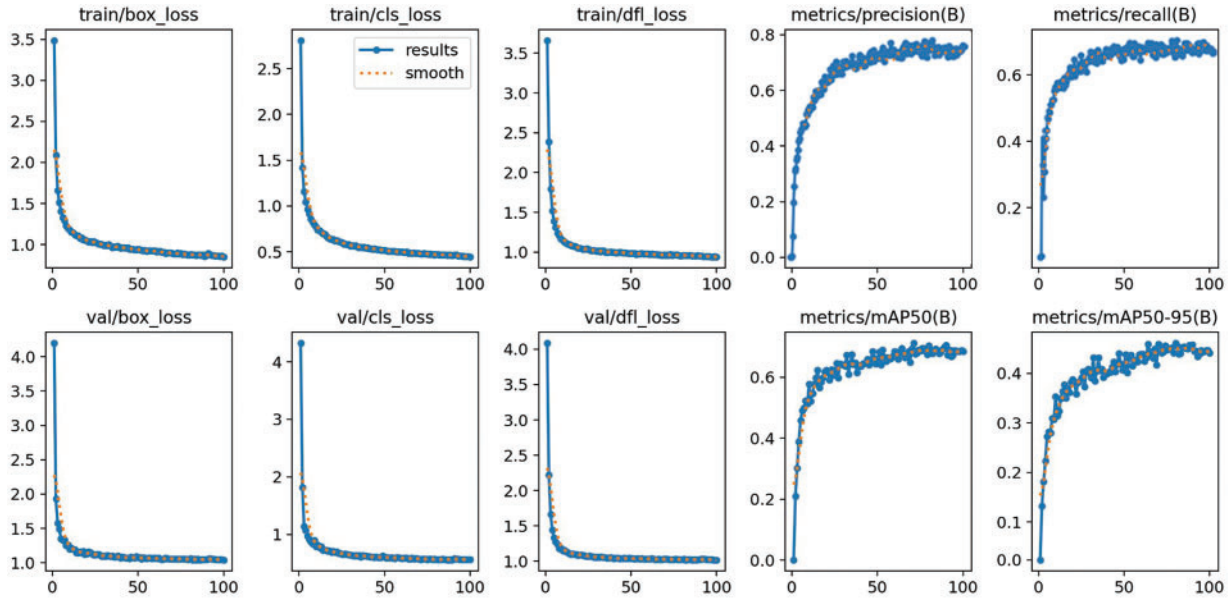

Figure 5: BES-Net model training evaluation results

4.6 Ablation Experiment

In order to verify the effectiveness of BoTNet, EMA and Slideloss, the improved methods were compared with ablation experiments. "√" indicates that the corresponding module is added, and "-" indicates that the corresponding module is not added. BES-Net is the model for adding all the above modules. The ablation experiment data are shown in Table 1.

Table 1: Comparison of network experiment results

BoTNet	EMA	SlideLoss	Precision	Recall	mAP50	mAP50-95	Model size/mb	Inference time/s
-	-	-	67.6%	61.4%	64.7%	37.2%	20.44	26.6
√	-	-	68.7%	68.9%	70.0%	43.5%	20.44	25.9
-	√	-	69.9%	71.3%	71.0%	42.2%	20.47	27.4
-	-	√	70.6%	69.4%	66.7%	39.3%	20.12	27.3
-	√	√	69.5%	68.4%	71.6%	44.1%	20.51	24.1
√	-	√	72.4%	71.2%	70.6%	44.5%	20.47	25.8
√	√	-	75.3%	73.9%	72.6%	47.1%	20.10	24.2
√	√	√	74.8%	73.1%	73.2%	47.5%	20.43	24.9

From the data, it can be observed that when employing the BoTNet module in the baseline network, the network's Precision, Recall, and mAP50-95 improved by 1.1%, 6.5%, and 5.3%, respectively, while the model's inference speed remained essentially unchanged. This validates the positive role of the BoTNet module in enhancing vehicle detection performance. Furthermore, by introducing the EMA attention mechanism based on these improvements, Precision increased by 5.4% and mAP50 improved by 1.8%. This indicates that incorporating the EMA attention mechanism can strengthen the model's feature extraction capability,

thereby improving detection accuracy. Finally, our modified loss function SlideLoss achieved enhanced Precision and mAP50-95 metrics while slightly accelerating feature extraction speed, with only a marginal decrease in Recall and almost no increase in computational load. After implementing SlideLoss, the network achieved optimal overall performance. In summary, the improved algorithm in this study not only achieves enhanced detection rates but also maintains balanced performance across other metrics.

The loss function change curve is shown in Fig. 6. When analyzing the loss images, we can observe that the BES-Net model shows the best performance throughout the training process. Compared with other variants, the loss curve of this model consistently shows lower values, indicating higher learning efficiency and faster convergence in terms of parameter optimization. Specifically, in the initial phase (0–50 cycles), the loss values of all models decreased rapidly, reflecting the occurrence of significant improvements at the beginning of learning. However, as we move beyond 50 cycles, the advantages of the BES-Net model become more apparent. Its losses continue to decline steadily. This trend of continuous loss reduction indicates that the model can maintain good generalization ability in the later training and effectively avoid overfitting phenomenon. In addition, it can also be seen from the figure that the BES-Net model maintains the lowest volatility throughout the training period, which means that its prediction results are more stable and reliable.

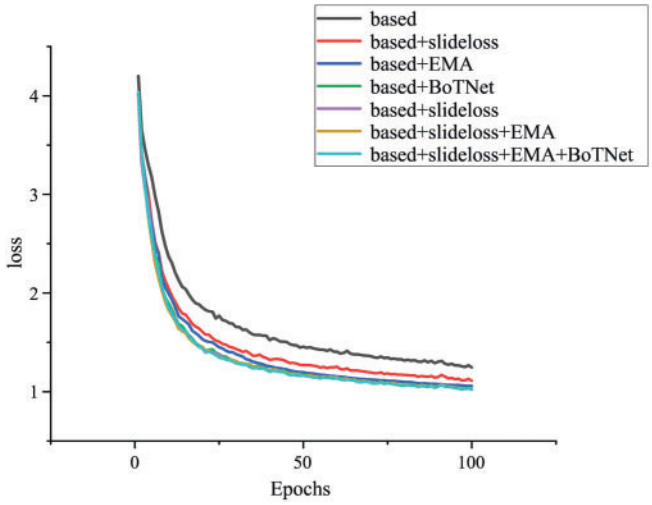

Figure 6: Change curve of ablation loss function

4.7 Model Generalization Verification

In order to more truly understand the actual working environment of vehicle detection, we added the motion blur effect on the basis of the self-built data set. The verification experiment results are shown in Table 2.

Table 2: Experimental results (add motion blur)

Model	Precision	Recall	mAP50	maP50-95	Inference time
BES-Net	74.3%	73.2%	72.9%	47.6%	20.3 ms

As can be seen from the experimental results in Table 3, the improved BES-Net network proposed in this paper still shows excellent performance when processing the data set with motion ambiguity added, reaching 74.3% accuracy. In addition, the inference speed of the model is very fast, the input image is processed and the output result takes only 20.3 ms. Fig. 7 shows the effect of the BES-Net network on the addition of a motion-fuzzy dataset.

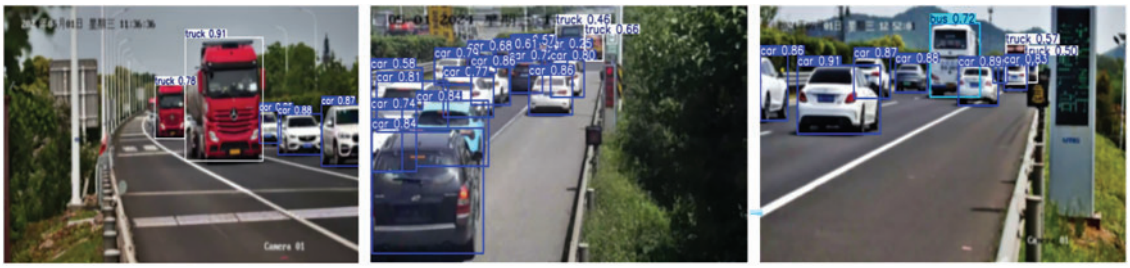

Figure 7: Part prediction effect of fuzzy processing data set

Table 3: NVIDIA Jetson Orin NX specific parameters

Parameter	Type
CPU	ARM Cortex-A78AE v8.2
AI performance	100 TOPS (INT 8)
Memory	16 G
GPU	NVIDIA Ampere
DL accelerator	2x NVDLA v2
Storage	128 G
Specification size	103 × 90 × 35 (mm)

4.8 BES-Net Edge Device Deployment

The combination of deep learning and embedded hardware has further promoted the popularity of smart hardware, especially in the field of intelligent transportation. However, the algorithms of road vehicle detection need to process a large amount of high-resolution image data, which puts higher requirements on the image processing capability and storage capacity of the equipment. And in practical applications, it may be affected by environmental lighting, noise and other factors, resulting in a decrease in positioning accuracy.

In order to realize the portable deployment of BES-Net, it can meet the requirements of intelligent transportation system. We deployed BES-Net, a vehicle detection network, on NVIDIA Jetson Orin NX equipment. The NVIDIA Jetson Orin NX is a high-performance, low-power embedded computing device. The specific parameters of the device are shown in Table 3.

The deployment of the BES-Net network on the NVIDIA Jetson Orin NX device leverages its powerful computing capabilities and efficient AI inference performance to enable real-time processing of complex deep learning models, thereby achieving high-precision target vehicle detection.

As can be seen from the data in Table 4, although the detection performance of BES-Net has decreased after it is deployed on NVIDIA Jetson Orin NX, it still maintains a high improvement compared with the original baseline network, and the detection speed has decreased a lot, which can ensure the real-time detection of vehicles.

Table 4: Experimental results

Model	Precision	Recall	mAP50	maP50-95	F1 score	Inference time
BES-Net	72.3%	71.2%	70.9%	45.6%	70.8	29.3 ms

4.9 Comparative Experiment

The proposed complex road vehicle detection module was compared in terms of detection performance with YOLOv5n, YOLOv7, YOLOv8n, and YOLOv8s. Fig. 8 shows a comparison of partial training results between other YOLO variants and BES-Net. It was observed that YOLOv5n failed to achieve satisfactory results, with lower detection performance across metrics such as accuracy and recall rate. It was found to have limitations in detecting vehicles in complex environments. YOLOv7 showed improvements across various metrics, but the test results produced significant errors, making the results unreliable. In comparison to the baseline model YOLOv8n, it was noted that, due to the use of simple contrastive learning (SimCLR), bidirectional feature pyramid network (BiFPN), and improved non-maximum suppression (NMS) algorithms, YOLOv8n performed well in feature extraction. However, it still missed or misidentified occluded vehicles. The improved BES-Net algorithm demonstrated significant performance improvements and showed strong adaptability to difficult and complex samples. Table 5 presents a comparison of the results between other YOLO variants and the proposed algorithm.

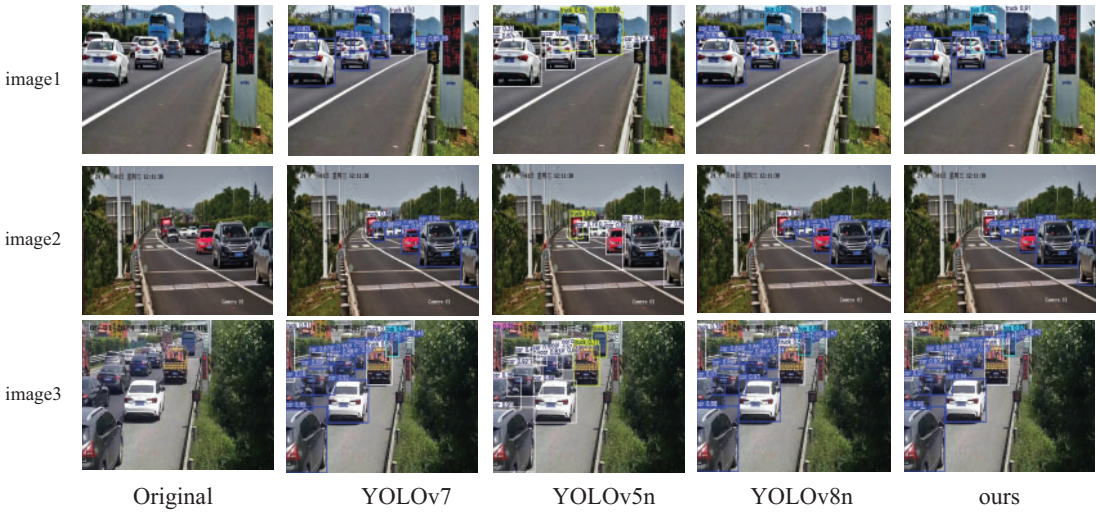

Figure 8: Comparison of training results between YOLO variants and BES-Net

Table 5: Comparison of network experiment results

Model	mAP50	mAP50-95	Precision	Recall	F1 Score
YOLOv5n	62.6%	38.2%	64.5%	59.8%	60.4%
YOLOv7	63.5%	36.5%	64.1%	60.1%	61.3%
YOLOv8n	64.7%	37.2%	67.6%	61.4%	60.8%
BES-Net	73.0%	47.5%	74.8%	73.1%	71.6%

4.10 Comparison with Other State-of-the-Art Models

The proposed complex road detection algorithm was compared with various other innovative methods on a self-built dataset. Tahir et al. [23] proposed a PVDM-YOLOv8l model for detecting vehicles and pedestrians under harsh conditions. They employed methods such as the convolutional block attention Module (CBAM) for feature refinement and a Swin Transformer for global feature extraction. While their model improved vehicle detection in adverse conditions, its mAP50 of 70.4% was lower than the proposed method. Wang et al. [24] introduced a YOLOv8-QSD model for detecting surrounding objects during vehicle movement. They enhanced the model using a diverse branch block (DBB), improving detection accuracy for high-speed vehicles. However, its generalization capability in complex vehicle environments remained limited, achieving an F1-score of 67.2% and mAP50-95 of 41.4%. Feng et al. [25] presented the ADWNet model, an improved version of YOLOv8 for object detection in autonomous driving. Their model incorporated SIOU loss and a multi-scale attention mechanism, enhancing precision (70.8%) and recall (71.4%). Nevertheless, its mAP50-95 (70.7%) fell below that of the proposed model. Safaldin et al. [26] enhanced YOLOv8's backbone network by introducing the Bi-PAN-FPN concept, significantly strengthening the algorithm's detection capability for moving objects. Although their model performed reasonably well with a precision of 71.1%, its mAP50 was only 39.2%. The results of these state-of-the-art models, compared with those of the proposed model, are summarized in Table 6.

Table 6: Comparison with other state-of-the-art methods

Model	mAP50	mAP50-95	Precision	Recall	F1 score
ADWNet	70.7%	42.5%	70.8%	71.4%	66.2%
YOLOV8-QSD	69.3%	41.4%	69.1%	71.1%	67.2%
PVDM-OLOv8	70.4%	43.6%	68.7%	69.3%	65.3%
PAN	71.2%	39.2%	69.6%	68.4%	65.8%
BES-Net	73.0%	47.5%	74.8%	73.1%	71.6%

5 Discussion

As the core artery of the national transportation network, China's expressways have faced increasingly severe traffic congestion in recent years. Especially during peak travel times such as holidays, in adverse weather conditions, or following unexpected accidents, certain sections of the roads often experience chain congestion due to a surge in traffic flow or traffic violations (such as random lane changes and slow driving blocking lanes), leading to a sharp decline in traffic efficiency and an increase in safety risks. Currently, Zhejiang Province has piloted AI video analysis systems, but their high computational costs and equipment maintenance fees make it difficult for small and medium-sized cities to adopt them widely. This paper introduces a multi-scale fusion algorithm, which provides a low-cost solution for intelligent road vehicle detection systems by deploying the algorithm into edge detection devices. The algorithm is based on YOLOv8, with improvements through three modules: BoTNet, EMA and slide loss. Compared to the baseline model, the accuracy, recall rate, and other metrics for vehicle detection have been improved. Additionally, our data augmentation strategy enhances the generalization ability of our algorithm, ensuring the input data's high quality and allowing the model to make more accurate predictions. The augmentation technique helps the system adapt to different environmental conditions, improving vehicle detection rates in complex road environments.

In real-world scenarios, the proposed model can integrate with low-cost sensors to serve as a prototype for future vehicle management and early warning systems, constructing a "perception-decision-regulation" closed-loop system covering the national road network. This not only reduces the economic losses of hundreds of billions of yuan caused by congestion each year but also lowers traffic accident rates and alleviates carbon emission pressure.

6 Conclusions

In this paper, we propose a network named BES-Net, and deploy the BES-Net network to NVIDIA Jetson Orin NX device, which realizes the convenient deployment of the network, Refines the practicality of the network, and achieves the accuracy of vehicle detection. Firstly, real road vehicles are relatively complex, so we enhance the long-distance dependence of the model by introducing a multi-head attention backbone network, and at the same time increase the detection speed of the network. Secondly, to solve the problem of complex road vehicles with obvious individual differences, we add the multi-scale fusion attention mechanism EMA to enrich the feature information and improve the expressiveness of the model. Finally, the slide loss function is introduced to solve the problem of the low detection rate of small samples. The experimental results demonstrate that BES-Net achieves competitive performance compared to conventional object detection models, with mAP50 reaching 73.2%, which is 8.5% higher than the baseline network. It is proved that the BES-Net network proposed in this paper can effectively detect vehicles in practical applications. In the future, we are committed to further improving the accuracy and real-time performance of the vehicle detection model and providing more efficient and reliable solutions in a wider range of application scenarios.

Acknowledgement: Not applicable.

Funding Statement: This research was funded by National Natural Science Foundation of China (No. 61961011), Innovation Project of Guangxi Graduate Education No. YCSW2025411.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Heng Wang, Jian-Hua Qin; data collection: Heng Wang; analysis and interpretation of results: Heng Wang; draft manuscript preparation: Heng Wang, Jian-Hua Qin. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

Abbreviations

ITS	Intelligent Transportation Systems
MAP	Mean Average Precision
EMA	Efficient Multi-Scale Attention Mechanism
CNN	Convolutional Neural Network
ML	Machine Learning
DL	Deep Learning
EVS	Electric Vehicles
IoT	Internet of Things
ECA	Efficient Channel Attention
SPP	Spatial Pooling Pyramid

PAN	Path Aggregation Network
MHSA	Multi-Head Attention
SGD	Stochastic Gradient Descent
CBAM	Convolutional Block Attention Module
DBB	Diverse Branch Block
BiFPN	Bidirectional Feature Pyramid Network
SimCLR	Simple Contrastive Learning

References

1. Lin Y, Wang P, Ma M. Intelligent transportation system (ITS): concept, challenge and opportunity. In: IEEE International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS); 2017 May 26–18; Beijing, China. p. 167–72. doi:10.1109/BigDataSecurity.2017.50.
2. Uijlings JRR, Van De Sande KEA, Gevers T, Smeulders AWM. Selective search for object recognition. Int J Comput VIS. 2013;13(2):154–71. doi:10.1007/s11263-013-0620-5.
3. Wang Z, Zhan J, Duan C, Guan X, Lu P, Yang K. A review of vehicle detection techniques for intelligent vehicles. IEEE Trans Neural Netw Learn Syst. 2023;34(8):3811–31. doi:10.1109/TNNLS.2021.3128968.
4. Chen C, Liu B, Wan S, Qiao P, Pei Q. An edge traffic flow detection scheme based on deep learning in an intelligent transportation system. IEEE Trams Intel Transp Syst. 2021;22(3):1840–52. doi:10.1109/TITS.2020.3025687.
5. Dalal N, Triggs B. Histograms of oriented gradients for human detection. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR); 2005 Jun 20-25; San Diego, CA, USA. p. 886–93. doi:10.1109/CVPR.2005.177.
6. Yu Z, Huang H, Chen W, Su Y, Liu Y, Wang X. YOLO-FaceV2: a scale and occlusion aware face detector. Pattern Recognit. 2024;155(10):110714. doi:10.1016/j.patcog.2024.110714.
7. Mazhar T, Asif RN, Malik MA, Nadeem MA, Haq I, Iqbal M, et al. Electric vehicle charging system in the smart grid using different machine learning methods. 2023;15(3):2603. doi:10.3390/su15032603.
8. Mehraban S, Yadav RK. Traffic engineering and quality of service in hybrid software defined networks. China Commun. 2024;21(2):96–121. doi:10.3390/sym15020513.
9. Ditta A, Ahmed MM, Mazhar T, Shahzad T, Alahmed Y, Hamam H. Number plate recognition smart parking management system using IoT. Measur Sensors. 2025;37(3):101409. doi:10.1016/j.measen.2024.101409.
10. Karangwa J, Liu J, Zeng Z. Vehicle detection for autonomous driving: a review of algorithms and datasets. IEEE Trans Intel Transp Syst. 2023;24(11):11568–94. doi:10.1109/TITS.2023.3292278.
11. Peng Y, Qin Y, Tang X, Zhang Z, Deng L. Survey on image and point-cloud fusion-based object detection in autonomous vehicles. IEEE Trans Intel Transp Syst. 2022;23(12):22772–89. doi:10.1109/TITS.2022.3206235.
12. Mao J, Shi S, Wang X, Li H. 3D object detection for autonomous driving: a comprehensive survey. Int J Comput Vis. 2023;131(8):1909–63. doi:10.1007/s11263-023-01790-1.
13. Singh N, Saini P, Shubham O, Awasthi R, Bharti A, Kumar N. Improved YOLOv5l for vehicle detection: an application to estimating traffic density and identifying over speeding vehicles on highway scenes. Multimed Tools Appl. 2024;83(2):5277–307. doi:10.1007/s11042-023-15520-9.
14. Song Y, Hong S, Hu C, He P, Tao L, Tie Z, et al. MEB-YOLO: an efficient vehicle detection method in complex traffic road scenes. Comput Mater Contin. 2023;75(3):5761–84. doi:10.32604/cmc.2023.038910.
15. Bochkovskiy A, Wang CY, Liao HYM. Yolov4: optimal speed and accuracy of object detection. arXiv:2004.10934. 2020.
16. Jocher G. YOLOv5 by ultralytics [computer software]. arXiv:2207.02696. 2020.
17. Li X, Qin Y, Wang F, Guo F, Yeow JT. Pitaya detection in orchards using the MobileNet-YOLO model. In: Proceedings of the Chinese Control Conference (CCC); 2020 Jul 27–29; Shenyang, China. p. 6274–8. doi:10.23919/CCC50068.2020.9189186.
18. Xu S, Wang X, Lv X, Chang Q, Cui C, Deng K, et al. PP-YOLOE: an evolved version of YOLO. arXiv:2203.16250. 2022.

19. Ding X, Zhang X, Ma N, Han J, Ding G, Sun J. RepVGG: making VGG-style convnets great again. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR); 2021 Jun 20–25; Nashville, TN, USA. p. 13733–42. doi:10.1109/CVPR46437.2021.01352.
20. Srinivas A, Lin TY, Parmar N, Shlens J, Abbeel P, Vaswani A. Bottleneck transformers for visual recognition. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR); 2021 Jun 20–25; Nashville, TN, USA. p. 16514–24. doi:10.1109/CVPR46437.2021.01625.
21. Ouyang D, He S, Zhan J, Guo H, Huang Z, Luo ML, et al. Efficient multi-scale attention module with cross-spatial learning. arXiv:2305.13563v2. 2023.
22. Russell BC, Torralba A, Murphy KP, Freeman WT. LabelMe: a database and web-based tool for image annotation. Int J Comput Vis. 2008;77(1–3):157–73. doi:10.1007/s11263-007-0090-8.
23. Tahir NUA, Zhang Z, Asim M, Iftikhar S, Abd El-Latif AA. PVDM-YOLOv8l: a solution for reliable pedestrian and vehicle detection in autonomous vehicles under adverse weather conditions. Multimed Tools Appl. 2024;1–26. doi:10.1007/s11042-024-20219-6.
24. Wang H, Liu C, Cai Y, Chen L, Li Y. YOLOv8-QSD: an improved small object detection algorithm for autonomous vehicles based on YOLOv8. IEEE T Instrum Meas. 2024;73:1–16. doi:10.1109/TIM.2024.3379090.
25. Feng X, Peng T, Qiao N, Li H, Chen Q, Zhang R, et al. ADWNet: an improved detector based on YOLOv8 for application in adverse weather for autonomous driving. IET Intel Transp Syst. 2024;18(10):1962–79. doi:10.1049/itr2.12566.
26. Safaldin M, Zaghden N, Mejdoub M. An improved YOLOv8 to detect moving objects. Proc IEEE Access. 2024;12:59752–806. doi:10.1109/ACCESS.2024.3393835.