IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 8 July 2024, accepted 24 July 2024, date of publication 29 July 2024, date of current version 8 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3435384

RESEARCH ARTICLE

A Lightweight Traffic Sign Detection Method With Improved YOLOv7-Tiny

XIAOBING CAO[1], YICEN XU[2], JIAWEI HE[1], JIAHUI LIU[3], AND YONGJIE WANG[3]
[1]School of Control Engineering, Wuxi Institute of Technology, Wuxi 214121, China
[2]School of Intelligent Equipment and Automotive Engineering, Wuxi Vocational Institute of Commerce, Wuxi 214153, China
[3]College of Computer and Control Engineering, Northeast Forestry University, Harbin 150040, China

Corresponding author: Xiaobing Cao (caoxb99@126.com)

This work was supported in part by the Qing Lan Project.

ABSTRACT As intelligent transportation systems continue to evolve, effective traffic sign detection is crucial for enhancing road safety and managing traffic congestion efficiently. This paper introduces the YOLOv7-tiny-RCA model, a novel lightweight approach designed specifically to address the dual challenges of high model complexity and diminished detection accuracy in long-distance scenarios commonly faced in traffic sign recognition. By integrating an improved ELAN-REP module into the YOLOv7-tiny backbone, our model significantly reduces computational complexity during the inference stage. Additionally, the incorporation of the CBAM attention mechanism enhances the model's capability to extract and fuse relevant information from traffic scenes more effectively. To further optimize performance, we replaced the traditional feature fusion network with an Asymptotic Feature Pyramid Network, which facilitates better interaction between different layers and reduces the overall computational burden. Our experimental results demonstrate that the YOLOv7-tiny-RCA model achieves a mean Average Precision of 81.03% and a parameter reduction from 6.13M to 4.99M, highlighting its efficiency and potential for deployment on edge devices. These significant improvements indicate that our model not only advances traffic sign detection technologies but also offers practical applications for modern intelligent transportation systems.

INDEX TERMS Asymptotic feature pyramid network, attention mechanism, object detection, RepConv, traffic sign detection.

I. INTRODUCTION

With the global advancement of autonomous driving and intelligent transportation systems, traffic sign detection has become a key research area. However, real road scenarios have intricate road conditions, so being able to quickly and accurately recognize traffic signs and provide drivers with important information that is difficult to capture directly with the naked eye has become the key to ensuring driving safety.

Traffic sign detection methods can be categorized into two types: methods based on traditional image processing and methods based on deep learning. In the traditional approach, traffic sign detection relies on analyzing its specific attributes. Since traffic signs are usually designed with strictly defined color patterns (e.g., red, blue, white) and specific shapes (e.g., circles, squares, and triangles). These properties help to separate them from the complex background environment. Therefore, traditional traffic sign detection methods are mainly classified into color-based, shape-based, and hybrid methods based on the combination of color and shape features. However, traditional traffic sign detection methods mainly rely on manually extracting features, but this approach is unable to deeply explore the deep semantic information of traffic signs, and performs poorly in terms of environmental adaptability and suffers from problems such as lack of robustness. In addition, this method is also limited in detection efficiency due to the complex computational process. In contrast, deep learning has the ability to extract features automatically, and achieves more efficient and accurate traffic sign detection by learning feature information through training.

The associate editor coordinating the review of this manuscript and approving it for publication was Kumaradevan Punithakumar.

With the rapid development in the field of deep learning, algorithms such as Faster R-CNN [1] and YOLO series have been widely used in the task of detecting road traffic signs. Li and Wang [2] designed a detection method that combines the Faster R-CNN framework and MobileNet structure, which is capable of adapting to different lighting and environmental conditions. Tabernik and Skocaj [3] improved the Mask R-CNN technique to recognize and detect large traffic signs, and the experimental results show that its error rate is no more than 3%. Based on Faster R-CNN, Pon et al. [4] performed global category classification and fine-grained classification through two-layer classification to effectively detect traffic signs and signals in different datasets. Although the CNN-based two-stage detection network excels in accuracy, it has some limitations in real-time due to high computational complexity. As a result, researchers are turning to single-stage networks for application scenarios that require higher real-time performance. Zhang et al. [5] designed their real-time Chinese traffic sign detection algorithm using YOLOv2, which optimizes the computational complexity by utilizing multiple 1×1 convolutional layers. Rajendran et al. [6] developed a YOLOv3-based traffic sign recognition approach that reduces the instability problem of detecting small-sized targets. Nevertheless, the detection accuracy of the method is still low and the real-time performance is still not up to the expected level. Chen et al. [7] developed a small traffic sign recognition technique called TSR-SA based on the YOLOv4 and YOLOv5 frameworks, and significantly improved the accuracy and robustness of the detection of small traffic signs through innovative data augmentation methods and model structural adjustments. Luo et al. [8] developed a YOLOv4-Ghost-based traffic sign detection algorithm, which optimized the network structure by integrating the GhostNet lightweight backbone network, and demonstrated excellent performance in terms of real-time and accuracy. In the 2021 study, Yang et al. [9] applied the YOLOv5 algorithm to the field of traffic sign detection. Better real-time performance was achieved by combining Mosaic data enhancement techniques and network parameter optimization. These studies improved the ability of single-stage networks to detect traffic signs by optimizing the network structure and other methods. However, these algorithms still face the problem of long detection process time due to the large number of parameters of the network model and high computational effort, which is difficult to meet the practical needs of lightweight traffic sign detection.

In order to ensure that the model can efficiently and accurately perform traffic sign detection under various edge device conditions, so as to effectively deal with various challenges in real traffic environments, this study takes the current mainstream lightweight real-time target detection algorithm YOLOv7-tiny as the baseline model, and proposes a new algorithm, YOLOv7-tiny-RCA, to reduce the number of model parameters and computation volume, while improving the detection accuracy, so as to better deploy on edge detection devices for traffic sign detection., computation, and can be better deployed on edge detection devices for traffic sign detection. The main contributions of our study are:

- In this study, we selected the TT100K [10] dataset. To address the problem of uneven distribution of categories in TT100K, we screened 45 categories with a sample size of more than 100, and expanded the self-constructed dataset by collecting road images containing traffic signs from some cities in Northeast China to constitute a dataset with a more balanced number of categories, and named it TT100K-B.
- In order to realize the lossless compression of the model, we propose an ELAN-REP module containing RepConv instead of the ELAN module in the YOLOv7-tiny backbone network, which draws on the idea of reparameterization to transform the multi-branch structure in the training stage into a single-branch structure in the inference stage, thus improving the inference speed of the model.
- To enhance the efficiency of our network in focusing on critical features while minimizing redundant information, we have integrated the Convolutional Block Attention Module (CBAM) into the feature layer of the YOLOv7-tiny's backbone network. This innovation significantly boosts the network's ability to characterize and process relevant features, thereby enhancing overall performance.
- Addressing the challenge of inadequate integration of high- and low-level semantic information within deep neural networks, we have employed a novel feature fusion approach called the Asymptotic Feature Pyramid Network (AFPN) within the YOLOv7-tiny's neck network. This technique not only elevates the model's accuracy and ability to capture detailed features in traffic sign detection but also reduces the number of model parameters and computational demands, substantially enhancing the model's efficiency on edge devices. Furthermore, we have replaced the conventional loss function with SIOU loss to improve the model's convergence and robustness, thereby optimizing YOLOv7-tiny for more effective traffic sign detection.

The structure of this paper is as follows: Section II introduces two-stage and one-stage detection algorithms, advancements in lightweight networks, and the detailed architecture of YOLOv7-tiny. Section III elaborates on the architectural enhancements of YOLOv7-tiny-RCA. Section IV details the experimental setup, datasets used for training and validation, and conducts ablation studies, comparisons with classical models, state-of-the-art (SOTA) comparisons, and result visualizations to demonstrate the effectiveness of the proposed method. Section V concludes the paper, discussing the limitations of the approach and future research directions.

II. RELATED WORK

A. SINGLE-STAGE AND TWO-STAGE DETECTORS

Deep learning-based target detection techniques are broadly categorized into two groups: first, two-stage detection algorithms such as Faster R-CNN [1] and Mask R-CNN [11]. Such algorithms perform target detection in two steps, where candidate frames are first generated from the input image, and then these regions are further processed, including regression to adjust the bounding box and classification. Another class of algorithms is the single-stage detection algorithms represented by SSD [12] and YOLO series. Unlike traditional methods, these methods realize fast end-to-end detection by treating target detection as a regression problem. Compared to single-stage algorithms, two-stage detection algorithms have higher detection accuracy, but it is difficult to meet the demand of real-time target detection because of their larger computation, higher complexity, and slower detection speeds.

Liang et al. [13] propose an improved Sparse R-CNN for traffic sign detection in autonomous vehicles. The framework incorporates self-adaptive augmentation and a new dataset, BCTSDB, enhancing detection accuracy in diverse conditions. The model addresses challenges in real-world applications, demonstrating significant improvements in detection robustness and accuracy. Wang et al. [14] develop a visual perception framework for intelligent transportation systems in the Metaverse, focusing on efficient data optimization and domain-adaptive learning. The framework, termed MITVF, enhances traffic object detection by leveraging virtual data for model training, significantly reducing costs and improving detection accuracy in complex traffic scenarios. They [15] introduce an efficient traffic sign detection system using a multi-scale approach and attention fusion. Their model, designed for vehicle-mounted systems, improves detection through an Attention Fusion Pyramid Network and a multi-head detection mechanism, achieving high accuracy and real-time performance with practical deployment capabilities on vehicle platforms.

B. YOLO SERIES NETWORK

YOLOv1 [16] has received a lot of attention for its fast detection speed since its introduction in 2016. It treats target detection as a regression task and predicts classification probabilities through regression analysis. Unlike the design of YOLOv1, YOLOv2 [17] removes the fully connected layers from the network so that the entire network structure consists only of convolutional and pooling layers.This improvement allows the network to be more flexible in handling images of different sizes.YOLOv2 also uses an odd-sized feature map design and updates the border position by calculating the coordinate offsets.YOLOv3 [18] Based on the Darknet-53 architecture, which is more complex than the Darknet-19 architecture of YOLOv2, and thus is able to better learn and represent complex features, it uses the structure of a feature pyramid network (FPN), which allows the network to learn and recognize targets of different sizes, effectively enhancing the detection of multi-scale targets. The YOLOv4 [19] algorithm has improved the network structure comprehensively by using CSPDarknet53 as the backbone network. The algorithm uses Spatial Pyramid Pooling (SPP) and Path Aggregation Network (PANet) to further process and fuse the features after feature extraction, which significantly improves the accuracy of traffic sign detection. The YOLOv5 [20] algorithm consists of a backbone network and a neck network. In the backbone network, the cross-stage partial network (CSP) improves the efficiency of feature extraction and effectively reduces the amount of computation. In addition, the backbone network contains a Spatial Pyramid Pooling Fast (SPPF) module, which is responsible for capturing the global information of the target. The neck network, on the other hand, employs PANet to aggregate features and generate a feature pyramid. This mechanism enhances the detection of targets at different scales, thus enabling the model to recognize the same target at different sizes and scales.

Currently, the YOLO family of algorithms has evolved to YOLOv7 [22]. The CBS module, Efficient layer aggregation networks (ELAN), and MP are integrated in the backbone network of YOLOv7. Among them, the CBS module combines a convolutional layer with a batch normalization layer and uses SiLU as an activation function to introduce nonlinear processing capability. The ELAN module, on the other hand, is designed with two branches to take on the tasks of channel transformation and feature extraction, which improves the generalization capability of the model by regulating the shortest and longest gradient paths. The MP module, on the other hand, implements image down-sampling through convolution and maximum pooling. YOLOv7 also uses MPConv, which draws on the advantages of both max-pooling and convolution to extract features efficiently, and combines the two to form a two-branch structure to capture richer feature information. The YOLOv7 architecture uses up-sampling in its neck module to fuse high-level and low-level feature information to facilitate the top-down flow of information.In addition, YOLOv7 adopts the SPPCSPC structure, which not only improves the processing speed and accuracy of the model by integrating the maximum pooling operation, but also enhances the model's ability to adapt to different resolution images. Finally, in the head module of the model, YOLOv7 corresponds to targets of large, medium and small scales through different scales of detection heads. Notably, in order to further improve the inference speed of the network and reduce the consumption of resources, YOLOv7 introduces RepConv in the detection head part, which enables it to improve the inference efficiency while keeping the detection accuracy, and significantly reduces the computational cost. However, in the YOLOv7-tiny version, its detection head uses standard convolution to replace RepConv in YOLOv7. In addition, to meet the needs of different devices, YOLOv7 authors design three types of basic models: YOLOv7, YOLOv7-W6, and YOLOv7-tiny. respectively, they are suitable for normal GPUs, cloud platform GPUs, and edge GPU hardware. In summary, the YOLOv7 algorithm

demonstrates its efficiency and accuracy in the real-time traffic sign detection task. Wang et al. [22] propose TSD-YOLO, an improved YOLOv8 for small traffic sign detection. It introduces the Space-to-Depth (SPD) module and Select Kernel (SK) attention mechanism to enhance detection accuracy in complex environments. The model shows significant improvements in detection performance on CSUST Chinese Traffic Sign Detection Benchmark (CCTSDB) and TT100K datasets.

C. LIGHTWEIGHT NETWORK

With the development of convolutional neural networks, neural networks have been widely used in industrial computer vision. However, in the early development stage of deep learning, the attention to the number of model parameters and the computational volume is not sufficient, which limits the application of the model in the field of real-time traffic sign detection. As a result, lightweight networks have emerged. Lightweight networks are mainly designed to reduce the number of parameters, computational size, and improve the speed of the network without sacrificing the accuracy of the model as much as possible, so as to achieve a balance between lightweight, real-time, and accuracy.

The SqueezeNet network [23], introduced in 2016, is one of the early representatives of lightweight networks, which achieves efficient compression of parameters by introducing the Fire module. Based on this, SqueezeNext [24] further improved the efficiency of the network by adding separation convolution. Subsequently, ShuffleNett [25] enhanced the information exchange between different channels by channel shuffling technique. By 2020, Han et al. proposed the GhostNet [26] lightweight network architecture. This architecture achieves more efficient feature extraction by reducing the number of model parameters and effectively handles information redundancy in the network. However, despite the success of the Ghost module in reducing the amount of computation, its ability to capture spatial information was reduced. Therefore, GhostNetV2 [28] introduces a Decoupled Fully Connected (DFC) based on a fully connected layer. This mechanism is not only less demanding on hardware, but also capable of capturing the dependencies between distant pixels, thus improving the inference speed of the model. Tang et al. [28] introduce ALODAD, an anchor-free lightweight object detector for autonomous driving. By integrating an attention scheme into the GhostNet backbone and using a novel IoU-aware head, ALODAD achieves high detection accuracy with lower computational costs. The model demonstrates robust performance in real-time traffic scenarios, outperforming several existing methods.

In addition, MobileNetV3 [29] is one of the lightweight networks in the MobileNet family, which follows the Deep Separable Convolution (DWConv) used in the MobileNetV1 [30] network and uses the inverse residual structure from MobileNetV2 [31], as well as introducing the SENet attention mechanism with lightweight improvements. These improvements help to reduce the loss of features in the extraction process and enhance the network's ability in capturing information.

D. YOLOV7-TINY NETWORK STRUCTURE

YOLOv7-tiny model is the lightest version of YOLOv7 series, with smaller number of model parameters and faster processing speed. Traffic sign detection faces complex road scenes in the real world, so it not only requires high real-time performance, but also has high requirements for accuracy. Currently, real-time target detection algorithms mostly need to be deployed on edge devices close to the user, such as cell phones and in-vehicle cameras. However, the computational power of these devices is limited to some extent compared to cloud platform GPUs. Therefore, based on lightweight considerations and the need for accuracy, YOLOv7-tiny, which has a smaller number of parameters and computation, is chosen as the benchmark model in this study.

The backbone feature extraction network of YOLOv7-tiny mainly consists of a CBL (Convolutional, Batch normalization, and Leaky ReLU) module, a concise version of the ELAN module and a maximum pooling module. Among them, the CBL module consists of a convolutional layer, a BN, and a Leaky ReLU activation function.The ELAN module mainly consists of VoVNet [32] and CSPNet [33], which optimizes the feature aggregation function while solving the problem of the model's difficulty in convergence after scaling. This ensures that the model maintains accurate performance even with fewer parameters and faster detection speeds, making it well suited for real-time traffic sign detection requirements and easy to deploy on edge devices such as in-vehicle cameras. In addition, YOLOv7-tiny eliminates the operation of connecting the two paths of max-pooling and convolution in the YOLOv7 version of MPConv, and instead uses only max-pooling for down-sampling. YOLOv7-tiny's necking network employs the SPPCSP module, which implements feature graph fusion of local and global features to not only enhance feature extraction but also provide richer feature graphs, as well as aggregating feature gradient information to reduce parameters while maintaining faster performance. YOLOv7-tiny employs the same YOLOv5 path-aggregated feature pyramid (PAFPN) architecture for feature aggregation, which combines FPN [34] and PANet [35] to achieve top-down fusion of low-resolution high-semantic information feature maps with high-resolution low-semantic information feature maps and top-down path augmentation to obtain localization information, such that each feature layer contains information from different layers.YOLOv7-tiny will take the feature maps with different scales generated after going through the process of feature extraction and multi-scale feature fusion in the Head section, and after further processing, the three feature maps with different sizes ($80 \times 80 \times 150$, $40 \times 40 \times 150$, and $20 \times 20 \times 150$) obtained correspond to the large, medium, and small target sizes, respectively. of the detected feature layers.

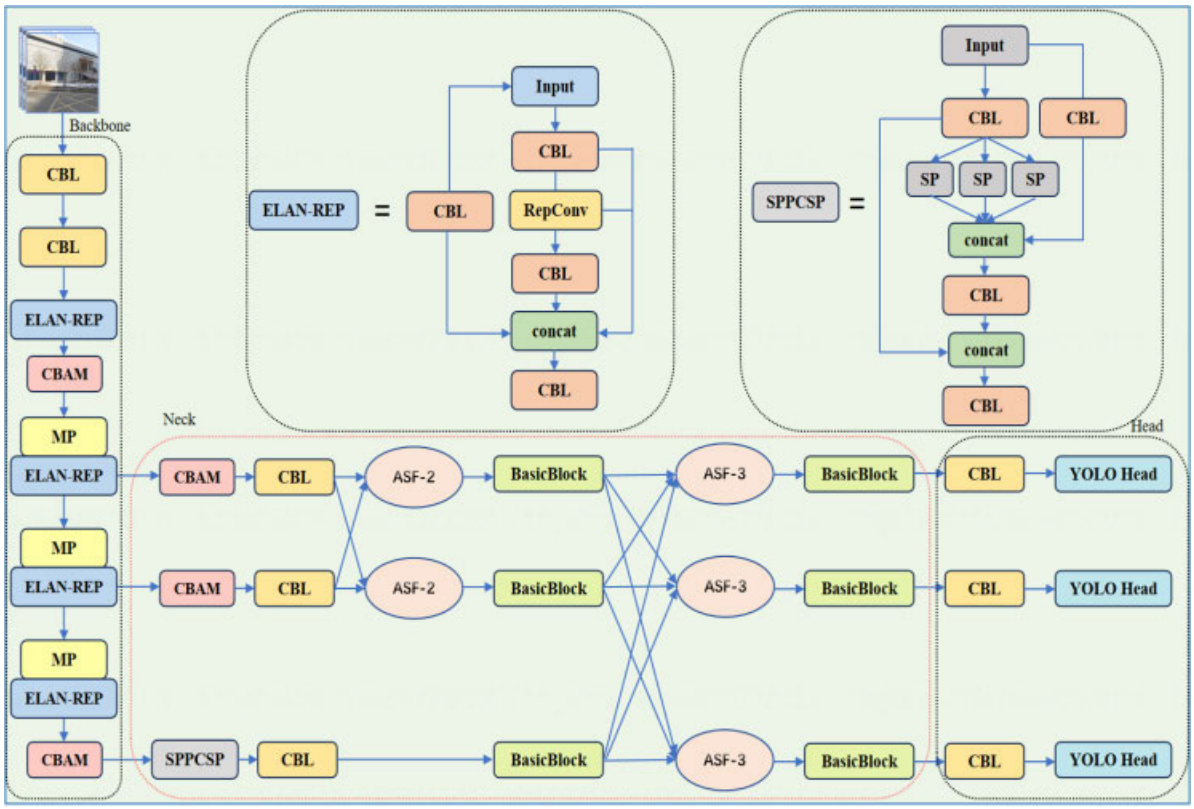


FIGURE 1. The architecture of our proposed YOLOv7-tiny-RCA.

III. MATERIALS AND METHOD

A. YOLOv7-TINY-RCA

In order to facilitate the deployment of the model on edge devices such as in-vehicle systems, this study makes improvements for YOLOv7-tiny to realize the lightweight requirement. Specifically, we first propose an ELAN-REP module containing RepConv instead of the ELAN module in the backbone network, which draws on the idea of reparameterization to transform the multi-branch structure in the training phase into a single-branch structure in the inference phase, thus improving the inference speed of the model. Subsequently, the CBAM attention mechanism is used in the feature layer of the backbone network to enable the network to reduce redundant information, focus on important features more effectively, and improve the network's representational capability and performance. In addition, the feature fusion technique AFPN is used to replace PANet in YOLOv7-tiny in order to adjust the feature fusion structure and improve the speed of model feature extraction. Finally, the loss of the original network is improved to enhance the convergence ability and robustness of the model through SIOU, which makes YOLOv7-tiny more suitable for traffic sign detection. The YOLOv7-tiny-RCA framework proposed in this paper is shown in Figure 1.

B. IMPROVED ELAN-REP

In 2021, Ding et al. [36] proposed a multi-branch network distribution architecture based on VGG [37]. This architecture takes advantage of the multi-branch structure to optimize the model during the training process and improves the efficiency of device memory usage and model inference speed through the single-branch structure in the inference phase. Based on this design concept, this paper uses RepConv to improve the performance of traffic sign detection while realizing complex training and simple inference of the model. The main idea is to optimize the network in the inference stage and fuse multiple computational modules into a single module, thus reducing the computational burden and memory requirements during inference. The reparameterization process consists of two main steps:

1) FUSION OF CONVOLUTIONAL AND BN LAYERS

In the inference stage, the fusion of the convolutional kernel with the BN layer and the fusion of the directly connected branches with the 3×3 and 1×1 convolutional layers are the two key points of the reparameterization. The formula for the fusion of the convolutional kernel with the BN layer can be expressed as:

The convolutional layer operation is specified as follows:

$$Conv\,(x) = B + W\,(x) \tag{1}$$

The BN layer operation is specified as follows:

$$\mathrm{BN}\,(\mathrm{x}) = \gamma * \frac{(\mathrm{x} - \mathrm{mean})}{\sqrt{\mathrm{var}}} + \beta \tag{2}$$

Combining the results of the convolution with the BN layer and substituting (1) into (2) yields (3):

$$BN\left(Conv\left(x\right)\right) = \gamma * \frac{\left(W\left(x\right) + B - mean\right)}{\sqrt{var}} + \beta \quad (3)$$

From the results, we can learn that after the fusion of the convolutional and BN layers, the new BN operation can be obtained by further organizing (4) as follows:

$$BN\left(Conv\left(x\right)\right) = \frac{W\left(x\right) * \gamma}{\sqrt{var}} + \left[\frac{\left(\left(B - mean\right) * \gamma\right)}{\sqrt{var}} + \beta\right] \quad (4)$$

2) MULTI-BRANCH FUSION IS ADOPTED

The re-parameterization technique uses a is a multi-branch composite structure containing 3×3, 1×1 convolutions as well as directly connected paths in the training phase. This structure is designed to allow the model to learn the features of the input data more fully, and to enhance the model's understanding of the input data by combining convolutional kernels of different sizes with directly connected paths. In order to improve efficiency and reduce model complexity during the inference phase, the three branches described above will be uniformly transformed into a single 3×3 convolution structure and fused together to form a single 3×3 convolution structure. This operation both reduces the model complexity and speeds up the inference of the model without sacrificing accuracy. This technique makes the model more suitable for operating in resource-constrained environments, such as real-time application scenarios in mobile devices or embedded systems, by reducing the amount of computation the model has to perform during inference. The multi-branch fusion process is specifically shown in Figure 2.

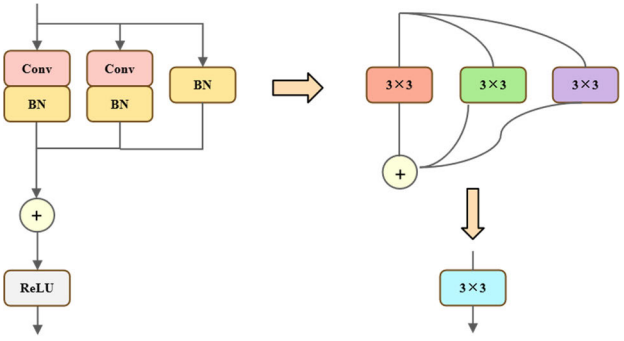


FIGURE 2. Schematic diagram of Multi-branch fusion process.

Therefore, in order to further improve the efficiency of YOLOv7-tiny network in the inference stage, this study adopts RepConv, which replaces the first 3×3 standard convolutional layer in the ELAN module of the original YOLOv7-tiny algorithm backbone network, as a way of improvement. Meanwhile, the improved ELAN module is named ELAN-REP module, and its specific structure is shown in Figure 3.

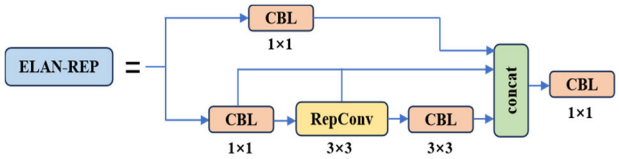


FIGURE 3. ELAN-REP module.

C. IMPROVED ATTENTION MODULE

CBAM [38] is a hybrid domain attention mechanism, which enables the model to associate features in channel and space more effectively through the combination of channel attention and spatial attention. As a result, CBAM can adaptively adjust the importance of features in the channel and spatial dimensions so that the network can focus more on key features and enhance its ability to capture features, thus improving the overall performance of the neural network. We opted for the CBAM due to its sequential attention mechanism that enhances feature representation by focusing on both channel and spatial dimensions. CBAM's integration into our YOLOv7-tiny model effectively improves its ability to localize and detect small or partially obscured traffic signs by emphasizing the most relevant features through channel-wise and spatial attention. This choice is substantiated by superior detection performance in complex backgrounds, distinguishing CBAM from other attention mechanisms that might only focus on one aspect of attention.

The schematic diagram of CBAM attention mechanism is specifically shown in Figure 4. Among them, the channel attention module mainly optimizes the model's adjustment of feature importance in different channels and transfers the captured feature information of different channels to a shared network to generate the channel attention graph. First, it performs global maximum pooling and global average pooling operations on the input feature maps. Subsequently, these pooling-operated feature maps are processed through a two-layer multilayer perceptron, which is then processed by a Sigmoid activation function to generate the final attention feature maps on the channel dimensions. The spatial attention module, on the other hand, focuses on focusing on the location information of the region of interest in the image and generates a spatial attention map based on the spatial relationship between the features. It follows the channel attention module and it takes the output feature map processed by the channel attention module as input. The two processed feature maps are then stitched together in the channel dimension by performing global maximum pooling and global average pooling operations on this feature map respectively. Next, the

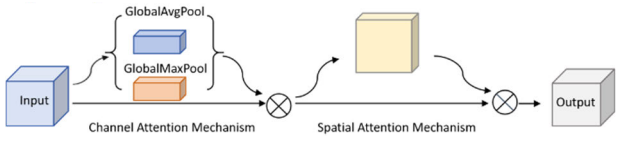


FIGURE 4. CBAM attention mechanism.

spliced feature map is downscaled by a convolutional layer of size 7×7 so that the output feature map has a channel number of 1. Finally, a Sigmoid activation function is applied to generate the spatial attention feature map. This combination design makes the channel and spatial attention mechanisms complement each other, thus improving the feature extraction capability of the model.

D. IMPROVED ASYMPTOTIC FEATURE PYRAMID NETWORK

In this study, a new feature fusion method, AFPN [39], is introduced in the neck network of YOLOv7-tiny to maintain the information integrity during the multilevel transmission process in order to enhance the extraction of target features of traffic signs. The specific structure of the AFPN used in this paper is shown in Figure 5. Where the black arrow represents the process of convolution and the blue arrow indicates the adaptive spatial fusion technique. In the initial stage of feature fusion, a feature set with multiple scales is formed by extracting the last layer of features from different levels of the backbone network at four scales, C2, C3, C4, and C5, as a way to achieve feature diversification at different scales. The AFPN firstly fuses the two neighboring low-level features, C2 and C3, and then gradually fuses the higher-level features upward, first C4 This hierarchical fusion effectively narrows the semantic gap between features at different levels, thus enhancing the effect of feature fusion.

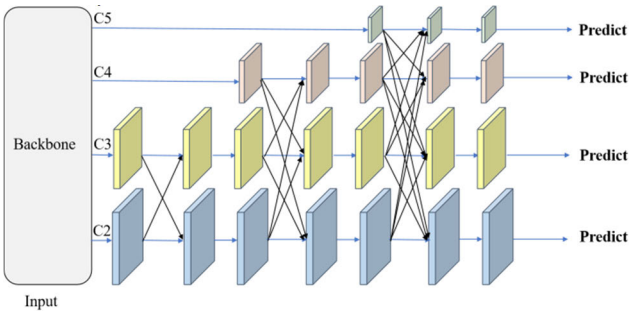


FIGURE 5. Asymptotic feature pyramid network architecture.

Our model incorporates a novel adaptation of the ASPP framework, specifically designed to improve the detection of traffic signs from varying distances and under different environmental conditions. By adjusting the atrous rates and optimizing the convolutional layers within ASPP, we have achieved finer granularity in feature extraction, particularly useful for recognizing small or partially occluded signs. This improved version of ASPP utilizes a combination of varying dilation rates to capture a broader context without losing resolution, enhancing the model's spatial resolution capabilities across different scales. The integration of these enhancements allows for a more detailed and accurate representation of features, significantly boosting the model's performance in complex scenes. Additionally, the modified ASPP contributes to a more efficient computation, reducing the overall computational burden while maintaining high detection accuracy.

As a result, in this study, the Neck network part of YOLOv7-tiny is improved and enhanced by introducing AFPN. Specifically, the YOLOv7-tiny-RCA algorithm first extracts the last layer of features from each feature layer of the improved backbone network to form a collection of features at different scales, C3, C4, and C5, respectively. Second, to achieve feature fusion, the bottomier C3 and C4 features are fed into the feature pyramid network first, and then the more complex top-most layer of features, C5, is added progressively. In summary, the process demonstrates the gradual integration of bottom, top, and top-level features in bottom-up feature extraction. This step-by-step feature fusion strategy effectively coordinates the semantic content between different layers of features in the AFPN, overcoming the challenges encountered when directly fusing features with significant semantic differences across layers.

In addition, in the process of multilevel feature fusion, different traffic sign targets at a certain spatial location may have potential conflicts due to the problem of information inconsistency between different levels. Therefore, the traditional element-level summation is not an effective method. To address this problem, this paper uses the adaptive spatial fusion operation ASFF [40] to mitigate this inconsistency. It consists of horizontal join, down-sampling, and up-sampling operations, which reinforce the importance of key levels by assigning different spatial weights to features at different levels, preserving the information useful for feature fusion. Figure 6 illustrates the specific operations of feature fusion for fusion at three different levels. The specific operations for adaptive spatial fusion are shown in (5) and (6). $x_{ij}^{n\to l}$ denotes the feature vector at position (i, j) in the transition from layer n to layer l. By performing adaptive spatial fusion on multiple layers of features, the resultant vector y_{ij}^{l} is obtained, which is composed of the linear combination of $x_{ij}^{1\to l} x_{ij}^{2\to l}$ and $x_{ij}^{3\to l}$.

$$y_{ij}^{l} = \alpha_{ij}^{l} \cdot x_{ij}^{1\to l} + \beta_{ij}^{l} \cdot x_{ij}^{2\to l} + \gamma_{ij}^{l} \cdot x_{ij}^{3\to l} \quad (5)$$

$$\alpha_{ij}^{l} + \beta_{ij}^{l} + \gamma_{ij}^{l} = 1 \quad (6)$$

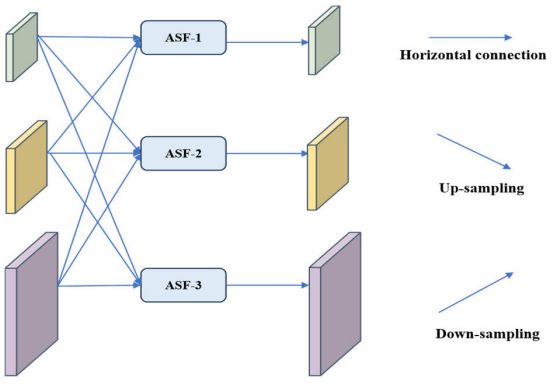


FIGURE 6. Adaptive spatial fusion operation.

where α_{ij}^{l}, β_{ij}^{l} and γ_{ij}^{l} denote the spatial weights of the three hierarchical features, which obey the constraints of (6). The adaptivity of the AFPN is achieved by assigning spatial weights $\left(\alpha_{ij}^{l}, \beta_{ij}^{l}, \gamma_{ij}^{l}\right)$ to the features at each level. This ensures that features from shallow to deep layers are optimized in the final fused feature vector y_{ij}^{l}. These weights are learned during the training process and their sum is 1 at any particular position (i, j) for the l layer, which is (6). For example, when shallow features need to be emphasized to obtain fine-grained spatial information, the network may tend to assign a larger weight to α_{ij}^{l}, while in contrast, β_{ij}^{l} and γ_{ij}^{l}, as the weights corresponding to deeper features, may be given smaller weights. Conversely, if the contextual information in the deep features is more important, the weights of β_{ij}^{l} and γ_{ij}^{l} will likely be boosted. These weighting parameters are dynamically adjusted during the training process based on the loss and backpropagated signals, accurately reflecting the importance of each feature level for the current detection task.

E. IMPROVED LOSS FUNCTION

In the baseline model YOLOv7-tiny, the CIOU loss was selected for bounding box regression loss calculation. Although CIOU considers the overlapping area of bounding boxes, centroid distance, and aspect ratio, its definition of aspect ratio is not explicit enough to accurately reflect the real difference between image width and height and their impact on model confidence. This leads to slower convergence and poorer stability during model training. Building on CIOU, SIOU [41] comprehensively evaluates the effects of distance, angle, and shape differences on bounding box regression. It incorporates the concept of angular difference between detection and prediction boxes, effectively promoting the prediction boxes to approach a parallel state with the ground truth more rapidly, thereby guiding the loss to optimize in the correct direction of convergence.

The angular cost is defined as (7), where Z_h represents the height difference between the centroids of the ground truth and prediction boxes, and σ represents the distance between the centroids of these boxes.

$$\Lambda = 1 - 2\times\sin^2\left(\arcsin\left(\frac{Z_h}{\sigma}\right) - \frac{\pi}{4}\right) \tag{7}$$

The distance cost is detailed from (8) to (11), where $b_{c_x}^{gt}$ and $b_{c_y}^{gt}$ are the centroid coordinates of the ground truth box, b_{c_x} and b_{c_y} are the centroid coordinates of the prediction box, and c_w and c_h are the width and height of the minimum bounding rectangle, respectively.

$$\Delta = \sum_{t=x,y}\left(1-e^{-y\rho_t}\right) \tag{8}$$

$$\rho_x = \left(\frac{b_{c_x}^{gt} - b_{c_x}}{c_w}\right)^2 \tag{9}$$

$$\rho_y = \left(\frac{b_{c_y}^{gt} - b_{c_y}}{c_h}\right)^2 \tag{10}$$

$$\gamma = 2 - \Lambda \tag{11}$$

The shape cost is defined in (12) to (14), where w and h are the width and height of the prediction box, w^{gt} and h^{gt} are the width and height of the ground truth box, and θ is a constant used to control the emphasis on shape loss. To further enhance the accuracy of bounding box shape matching and to constrain their reasonable movement in space, this study sets θ to 1.

$$\Omega = \sum_{t=w,h}\left(1-e^{-wt}\right)^{\theta} \tag{12}$$

$$\omega_w = \frac{\left|w - w^{gt}\right|}{\max\left(w, w^{gt}\right)} \tag{13}$$

$$\omega_h = \frac{\left|h - h^{gt}\right|}{\max\left(h, h^{gt}\right)} \tag{14}$$

In conclusion, the SIOU loss, which considers angular, distance, and shape costs, is specifically defined as shown in (15).

$$L_{SIOU} = 1 - IOU + \frac{+\Omega}{2} \tag{15}$$

This study employs SIOU as the loss function during the model training process. It is particularly suitable for multi-scale object detection tasks and assists in enhancing the detection performance of small objects, thereby improving the model's convergence performance.

IV. RESULTS AND ANALYSIS

A. EXPERIMENT SETTINGS

This study ultimately selected the category-rich Chinese traffic sign dataset TT100K to enhance the improved model's performance in multi-class scenarios. The TT100K dataset comprises approximately 100,000 panoramic images collected from multiple regions across five different cities in China, with about 10,000 images containing traffic signs. Each image is segmented into high-resolution sizes of 2048×2048, containing over 200 classes of traffic signs.

Nevertheless, the official TT100K dataset presents a significantly skewed class distribution, with merely 45 classes containing over 100 samples each. Hence, to align with real-world requirements and boost model accuracy, we excluded classes with fewer than 100 samples from the initial dataset, maintaining only the 45 classes that surpassed this threshold to guarantee the quality of the data. Subsequently, for the 20 categories with fewer samples among the 45 classes, the dataset was supplemented through a custom-built dataset to achieve data balance among classes. The custom-expanded dataset contains 1,834 images. Reflecting the structure of this dataset, we further refined the original TT100K dataset through meticulous processes such as filtering by category, cleansing data, and eliminating corrupted images, resulting in a curated dataset of 11,004 images. These were then segmented into training, testing,

and validation groups following an 8:1:1 distribution ratio. Specifically, the training set contains 8,802 images, while the testing and validation sets each contain 1,101 images. The final dataset, named "TT100K-B," includes 45 classes with a more balanced distribution among the classes. Additionally, to evaluate the model's generalization ability, this paper also introduces the CCTSDB in the comparative experiments. After data cleaning, this dataset retains 19,356 images for validation. Some of the dataset images are shown in Figure 7.

FIGURE 7. Partial images of the TT100K-B dataset.

The training process of this experiment was conducted on Ubuntu 18.04 with CUDA 11.7, using an NVIDIA Tesla V100S-PCIE GPU. The experimental code was primarily written and tested using the Pytorch framework and Python language. During model training, the input image size was set to 640 × 640. The model's batch size was set at 64, with data iteration training epochs set to 300. An early stop mechanism is used during the training process to set the training loss to be regulated early if it does not continue to decrease within 30 epochs. To optimize the algorithm, the experiment utilized the Adam optimizer, employing a cosine annealing strategy for learning rate adjustments with an initial learning rate set to 0.01. Additionally, this experiment utilized COCO pretrained weights from yolov7-tiny.pt for initialization.

We selected FPS, Params, GFLOPS, Precision, Recall, and mean Average Precision (mAP) as performance metrics, with some of the formulas presented as follows:

$$Precision = \frac{TP}{TP + FP} \tag{16}$$

$$Recall = \frac{TP}{TP + FN} \tag{17}$$

$$AP = \int_0^1 P(r)\, dr \tag{18}$$

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i \tag{19}$$

where TP refers to the number of target bounding boxes correctly identified as positive, which are the traffic signs that are correctly labeled. FP represents the number of detection boxes that erroneously identify negative class objects as targets, such as mistaking background or other non-traffic sign objects as traffic signs. FN denotes the number of actual targets not detected, representing the missed traffic signs. TN refers to the correct detection of negative classes. Given the complexity of the image background in traffic sign detection tasks and the presence of a large number of non-target bounding boxes, TN is not considered.

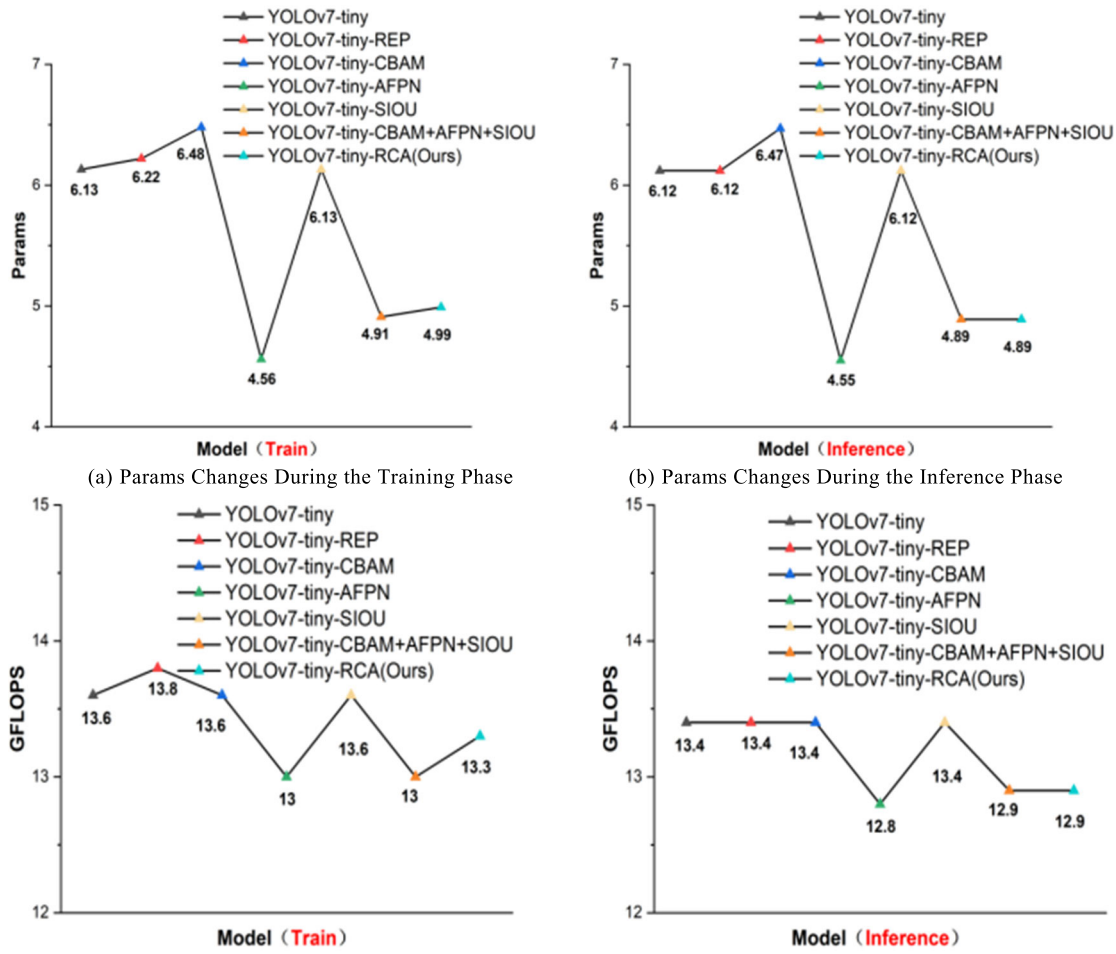
B. ABLATION STUDY

To comprehensively verify the performance of each module within the YOLOv7-tiny-RCA algorithm, this paper conducted ablation studies on the YOLOv7-tiny-RCA algorithm and its four improved modules under the same experimental conditions. The specific results are shown in Table 1. The experimental outcomes demonstrate that, under the TT100K dataset, compared to the baseline model YOLOv7-tiny, each module independently contributes to the enhancement of various performance metrics of the model when tested in isolation.

We first discuss the proposed improvement in the algorithm YOLOv7-tiny-REP, where the ELAN-REP module introduced in this paper replaces the original ELAN module in the backbone network of YOLOv7-tiny. To visually compare the differences in effects before and after the introduction of the ELAN-REP module, Figure 8 illustrates the changes in parameters and computational load during the training and inference stages. Notably, the figure also includes configurations of YOLOv7-tiny combined with CBAM, AFPN, and SIOU for comparative analysis, highlighting the role of the ELAN-REP module in the comprehensive improvement of the algorithm.

From Figure 8, it can be observed that during the training phase, the Params and GFLOPS of the YOLOv7-tiny-RCA model slightly increase. However, in the inference phase, compared to YOLOv7-tiny combined with CBAM, AFPN, and SIOU, the introduction of the ELAN-REP module in the comprehensive improved module YOLOv7-tiny-RCA does not actually increase the Params and GFLOPS. This indicates that although there is a slight increase in parameters and GFLOPS during the training phase, this increase can be effectively mitigated and fused through reparameterization techniques in the inference stage, without affecting the final model performance. Additionally, YOLOv7-tiny-REP shows a 1.14% improvement in mAP over YOLOv7-tiny, with FPS increasing from 110 to 115. These experimental results demonstrate that the introduction of RepConv achieves improved accuracy and FPS in the inference stage, without changing the parameter count or computational load, thereby validating its practical application value in enhancing model inference performance.

In this study, we utilized the data-class balanced TT100K-B dataset to compare the performance differences between the baseline model YOLOv7-tiny and the improved model YOLOv7-tiny-REP. Through the visual comparison in Figure 9, both models exhibited high accuracy in identifying traffic signs at greater distances, such as w59 and p5.



(a) Params Changes During the Training Phase
(b) Params Changes During the Inference Phase
(c) GFLOPS Changes During the Training Phase
(d) GFLOPS Changes During the Inference Phase

FIGURE 8. Diagram of Params and GFLOPS changes during training and inference stages across different models.

(a) YOLOv7-tiny Detection Results
(b) YOLOv7-tiny-REP Detection Results

FIGURE 9. YOLOv7-tiny-REP comparison of experimental results.

Particularly, YOLOv7-tiny-REP showed an increase in confidence scores of 0.04 and 0.03, respectively, compared to the baseline model YOLOv7-tiny. Furthermore, Figure 9 also demonstrated the robustness of the improved model in complex scenarios, where it was able to more accurately locate and recognize distant traffic signs. These results validate the effectiveness of the ELAN-REP feature extraction module in enhancing the performance of traffic sign detection and classification, highlighting its potential value in practical applications.

Secondly, we discuss the improved algorithm YOLOv7-tiny-CBAM, which incorporates the CBAM for mixed-domain attention in YOLOv7-tiny. According to Table 1, compared to the baseline model YOLOv7-tiny, the mAP of YOLOv7-tiny-CBAM has increased from 77.12% to 77.87%, indicating better performance in extracting semantic feature information. The visual analysis in Figure 10 further demonstrates the powerful recognition capabilities of YOLOv7-tiny-CBAM in handling complex scenes with multiple small traffic signs. Compared to the original model, YOLOv7-tiny-CBAM not only accurately identified the previously undetected pne signs but also detected w57 and i4 signs with higher confidence. These results prove that the CBAM module enhances the model's detection accuracy for small and complex targets, which is of significant value for improving the accuracy of traffic sign detection.

Subsequently, we also discuss the improvements to the YOLOv7-tiny model through the introduction of the AFPN.

(a) YOLOv7-tiny Detection Results (b) YOLOv7-tiny-CBAM Detection Results

FIGURE 10. YOLOv7-tiny-CBAM comparison of experimental results.

(a) YOLOv7-tiny Detection Results (b) YOLOv7-tiny-SIOU Detection Results

FIGURE 12. YOLOv7-tiny-SIOU comparison of experimental results.

As indicated by Table 1, compared to YOLOv7-tiny, the YOLOv7-tiny-AFPN model increased its mAP from 77.12% to 78.75%, achieving an improvement of 1.63 percentage points. At the same time, the model's parameter count and computational load were effectively reduced from 6.13M and 13.6G to 4.56M and 13.0G, respectively, significantly compressing the model size for deployment on resource-constrained edge devices. Although there was a slight decrease in FPS, the overall enhancement in performance metrics, as well as increases in precision and recall, fully validate the advantages of using AFPN to replace the original feature fusion network of YOLOv7-tiny in terms of feature extraction. Figure 11 demonstrates the practical application effects of AFPN through comparative experimental results. The comparison shows that the YOLOv7-tiny-AFPN algorithm can accurately detect and locate the categories w13 and pl20, with confidence levels increasing by 0.04 and 0.05, respectively, compared to the YOLOv7-tiny algorithm. These experimental results confirm the benefits of incorporating AFPN, enhancing the YOLOv7-tiny's ability to extract fine detail features. It also highlights the model's improved recognition performance for small targets in practical applications, verifying the effective enhancement of the original network by AFPN.

(a) YOLOv7-tiny Detection Results (b) YOLOv7-tiny-AFPN Detection Results

FIGURE 11. YOLOv7-tiny-AFPN comparison of experimental results.

Finally, the original CIOU loss in the YOLOv7-tiny algorithm was replaced with the SIOU loss as an improvement. According to the data in Table 1, compared to the baseline model YOLOv7-tiny, YOLOv7-tiny-SIOU increased its mAP to 78.05%, demonstrating the advantages of SIOU in enhancing the accuracy of locating occluded traffic signs. Moreover, the detection results comparison in Figure 12 further illustrates the benefits of SIOU. It was observed that both algorithms could accurately identify traffic signs of categories pl60 and p11 in complex scenes with partial occlusions, multiple traffic signs, and targets at a distance. However, compared to the YOLOv7-tiny algorithm, YOLOv7-tiny-SIOU showed an increase in confidence scores by 0.01 and 0.02, respectively. Additionally, YOLOv7-tiny experienced missed detections of the pn category traffic signs ahead, while YOLOv7-tiny-SIOU successfully avoided this issue. These results indicate that in scenarios with partial occlusions, smaller traffic signs, and greater distances, the YOLOv7-tiny-SIOU algorithm demonstrates superior recognition and localization capabilities.

SIOU is chosen over CIOU for its enhanced ability to handle skewed and rotated bounding boxes by incorporating scale variance and rotational adjustments directly into the loss function. This adjustment enhances the detection precision in complex scenarios, making SIOU particularly effective for traffic sign detection where objects may not align perfectly horizontal or vertical. Empirical tests comparing SIOU with CIOU have demonstrated a consistent improvement in detection accuracy, particularly in challenging conditions, validating the integration of SIOU into the YOLOv7-tiny-RCA model alongside other architectural enhancements. This synergy improves the overall model performance in real-world applications.

As shown in Table 1, the proposed lightweight model YOLOv7-tiny-RCA improves upon the baseline model YOLOv7-tiny in Precision, Recall, and mAP by 4.12%, 3.37%, and 3.91% respectively, demonstrating its advantages in capturing traffic sign image targets and effectively reducing the risks of overlooking critical information and missing detections. Additionally, the model's parameter count and computational complexity have been effectively controlled, reduced to 4.99M and 13.3 GFLOPS, respectively. Further, according to the schematic changes in Figure 8, the parameter count of the YOLOv7-tiny-RCA model in the inference

TABLE 1. Comparison of the ablation study results for the improved models on the TT100K-B dataset.

Model	Precision%	Recall%	mAP@0.5%	FPS	Pramas(M)	GFLOPS
YOLOv7-tiny（Baseline）	78.64	76.76	77.12	110	6.13	13.6
YOLOv7-tiny-REP	79.23	77.30	78.26	**115**	6.22	13.8
YOLOv7-tiny-CBAM	80.45	76.47	77.87	107	6.48	13.6
YOLOv7-tiny-AFPN	80.60	78.61	78.75	103	**4.56**	**13.0**
YOLOv7-tiny-SIOU	79.05	77.84	78.05	110	6.13	13.6
YOLOv7-tiny-REP+CBAM	80.85	76.98	78.43	112	6.57	13.8
YOLOv7-tiny-REP+CBAM+AFPN	82.50	79.07	80.31	104	4.99	13.3
YOLOv7-tiny-RCA（Ours）	**82.76**	**80.13**	**81.03**	104	4.99	13.3

(a) YOLOv7-tiny Detection Results

(b) YOLOv7-tiny-RCA Detection Results

FIGURE 13. YOLOv7-tiny-RCA comparison of experimental results.

phase is further reduced to 4.89M, and the computational complexity is decreased to 12.9 GFLOPS. The model also maintains a high running speed, with an FPS of 104. Moreover, the visual analysis in Figure 13 further demonstrates the performance and generalization ability of the YOLOv7-tiny and YOLOv7-tiny-RCA algorithms in real-world application scenarios.Comparison of detection results reveals that in scenarios containing multiple traffic signs, YOLOv7-tiny mistakenly identified the pl30 traffic sign as pl50, while YOLOv7-tiny-RCA not only accurately recognized pl30 and pn categories but also increased the confidence score for pl30 to 0.83. This experimental outcome confirms that in complex road environments, YOLOv7-tiny-RCA possesses higher stability and detection capabilities.

In summary, the improvements proposed in this study for the YOLOv7-tiny algorithm are effective. The experimental results validate the proposed enhancement strategies in improving detection accuracy while maintaining computational efficiency and real-time performance, fully demonstrating that YOLOv7-tiny-RCA offers greater potential for deployment on computationally constrained edge devices.

C. COMPARISON OF LIGHTWEIGHT CONVOLUTION AND NETWORK

This section compares the YOLOv7-tiny-AFPN with classic lightweight networks such as MobileNetv3 and GhostNetv2. Considering that replacing the neck or head parts with lightweight models could disrupt the original network's feature fusion structure and degrade detection performance, this paper employs lightweight models like MobileNetv3 and GhostNetv2 to replace the backbone of YOLOv7-tiny. Additionally, to assess the effectiveness of the YOLOv7-tiny-REP module, this experiment also compares it against an improvement strategy that involves replacing all the standard convolutions in the four ELAN modules of the backbone network with lightweight convolutions, specifically DWConv. The experimental results are shown in Table 2.

Table 2 shows the performance of various models in terms of mAP@0.5%, FPS, Params, and GFLOPS. The experimental results indicate that compared to the baseline model YOLOv7-tiny, lightweight networks such as MobileNetv3 and GhostNetv2 effectively reduce the model's parameters and computational load, but their mAP values decreased by 4.65% and 1.81% respectively. This decline may be attributed to the simplified network architectures, which compromise the ability to capture subtle features, resulting in a loss of detection accuracy. In contrast, YOLOv7-tiny-AFPN more effectively facilitates feature extraction in both shallow and deep layers of the network. The use of AFPN not only reduces the model's parameters and computational load but also leads to an improvement in the mAP value of the YOLOv7-tiny-AFPN model, reaching 78.75%. This structure provides YOLOv7-tiny-AFPN with better adaptability and robustness in complex environments, thereby achieving a significant increase in mAP. Therefore, compared to other lightweight networks, the introduction of the new feature fusion network AFPN shows better performance in terms of lightness, speed, and accuracy.

Furthermore, experimental results indicate that in the YOLOv7-tiny-DwConv model, despite a significant reduction in the number of parameters and computational complexity, and an improvement in FPS, making the model more suitable for resource-constrained devices, there is a notable decrease in detection performance. Compared to YOLOv7-tiny, the mAP@0.5% has decreased by 2.31%. This may be due to a lack of sufficient representational capacity to maintain the extraction of complex features. Therefore, this paper selects YOLOv7-tiny-REP as an improvement strategy. Compared to YOLOv7-tiny, the YOLOv7-tiny-REP algorithm increases the model's mAP from 77.12% to 78.26% in the inference phase without adding any Params or GFLOPS. By enhancing real-time performance, it effectively improves

TABLE 2. Comparison of experimental results for different lightweight convolutions and networks on the TT100K-B dataset.

Model	mAP@0.5%	FPS	Pramas(M)	GFLOPS
YOLOv7-tiny (Baseline)	77.12	110	6.13	13.6
YOLOv7-tiny-MobileNetv3	72.47	103	7.02	8.4
YOLOv7-tiny-GhostNetv2	75.31	91	6.28	6.0
YOLOv7-tiny-AFPN	**78.75**	**103**	**4.56**	**13.0**
YOLOv7-tiny-DwConv	74.81	113	4.40	9.30
YOLOv7-tiny-REP	**78.26**	**115**	**6.22**	**13.8**

TABLE 3. Comparison of experimental results with classic algorithms.

Model	Precision%	Recall%	mAP@0.5%	FPS	Pramas(M)	GFLOPS
YOLOv4-tiny	71.17	61.95	66.42	89	6.16	14.1
YOLOv5s	78.21	73.28	75.73	83	7.14	16.5
YOLOv7	85.5	80.8	87.3	62	37.3	105.7
YOLOv7-tiny	78.64	75.76	77.12	**110**	6.13	13.6
YOLOv7-tiny-RCA（Ours）	**82.76**	**80.13**	**81.03**	104	**4.99**	**13.3**

the model's detection accuracy, achieving complex training and straightforward inference.

D. COMPARISON OF BASELINE

To further validate the performance of YOLOv7-tiny-RCA in the field of traffic sign detection, this study conducted a comprehensive comparison of the algorithm with several classic object detection algorithms on the TT100K-B dataset. These include YOLOv4-tiny, YOLOv5s, YOLOv7, and the baseline model used in this research, YOLOv7-tiny. All experiments were conducted using the same configuration of equipment, dataset, data preprocessing, and augmentation strategies, and ensured a consistent training to test set ratio to guarantee fairness. After 300 iterations, the best results were selected for evaluation. Ultimately, Figure 14 presents the performance comparison of different models on the mAP@0.5% metric. Table 3 shows the specific comparative experimental results.

This experiment conducted a visual assessment of the improvements made by the YOLOv7-tiny-RCA algorithm, focusing on changes in the mAP@0.5% metric during the training process. As shown in Figure 14, YOLOv7-tiny-RCA exhibited a noticeable increase in mAP values during the initial 150 epochs. After 300 epochs of training, the mAP value of YOLOv7-tiny-RCA reached 0.8103, significantly higher than those of YOLOv7-tiny, YOLOv4-tiny, and YOLOv5s. This indicates that compared to YOLOv4-tiny and YOLOv5s, YOLOv7-tiny-RCA incorporates more advanced convolution and feature fusion technologies, making it more adept at handling traffic signs in small sizes and complex backgrounds. However, despite the higher mAP value of the YOLOv7 model, as shown in Table 3, its Params and GFLOPS are much higher than the lightweight algorithm

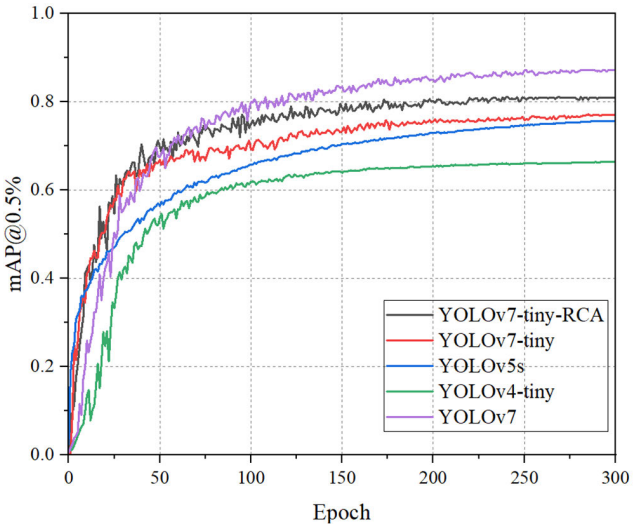


FIGURE 14. Comparison of mAP across different models.

YOLOv7-tiny-RCA proposed in this paper, making it less suitable for computationally constrained edge devices.

The comparative experimental results presented in Table 3 indicate that under the same experimental conditions, the YOLOv7-tiny-RCA proposed in this paper shows improvement in all metrics. Specifically, the YOLOv7-tiny-RCA algorithm has only 4.99M parameters, whereas the YOLOv7 model has a much higher parameter count of 37.3M. Additionally, the GFLOPS of YOLOv7-tiny-RCA is 13.3, significantly lower than other algorithmic models, demonstrating its ability to significantly reduce computational resource demands while maintaining performance. These

TABLE 4. Comparison of experimental results with advanced algorithms.

Model	mAP@0.5%	FPS	Pramas(M)	GFLOPS
Jia et al.	71.31	111	5.14	11.6
Qian el al.	76.48	79	5.73	46.5
Sharma el al.	64.90	**129**	9.51	36.2
Baseline model	77.12	110	6.13	13.6
YOLOv7-tiny-RCA（Ours）	**81.03**	104	**4.99**	**13.3**

results highlight the efficiency of the YOLOv7-tiny-RCA network structure, effectively minimizing unnecessary redundancy in parameters, which is particularly important in computationally constrained environments.

In terms of speed, YOLOv7-tiny-RCA achieves an FPS of 104, which is significantly higher than YOLOv7, YOLOv4-tiny, and YOLOv5s. Although it is a decrease of 6 FPS compared to the baseline model YOLOv7-tiny, this slight reduction does not impact its ability to meet the requirements of real-time traffic sign detection tasks. This indicates that YOLOv7-tiny-RCA, while achieving high accuracy, also effectively balances processing speed, making it suitable for applications that require real-time and rapid response.

On the mAP@0.5% metric, YOLOv7-tiny-RCA achieves 81.03%, significantly outperforming YOLOv4-tiny at 66.42%, YOLOv5s at 75.73%, and the baseline model YOLOv7-tiny at 77.12%. This performance improvement reflects the enhancements made by YOLOv7-tiny-RCA in feature extraction, background noise suppression, and target localization, particularly in complex or changing traffic environments where it can more accurately identify and categorize various traffic signs. Therefore, YOLOv7-tiny-RCA not only achieves increased accuracy and computational efficiency while maintaining real-time capabilities but also its lower resource consumption makes it especially suitable for deployment on low-power edge devices.

E. COMPARISON OF SOTA

To validate the advancement of the YOLOv7-tiny-RCA algorithm in the field of traffic sign detection, we extended our comparison to include other traffic sign detection methods on the TT100K-B dataset. Table 4 presents the comparative experimental results of the YOLOv7-tiny-RCA method against other traffic sign detection methods on the TT100K-B dataset. As demonstrated by the experimental data in Table 4, the YOLOv7-tiny-RCA method proposed in this paper shows a significant improvement in mAP compared to the traffic sign detection method improved from YOLOv7-tiny by Jia et al. [42] Compared to the baseline model YOLOv7-tiny, the mAP improved by 9.72%. Although the TSDet method, proposed by Qian et al. [43], has a higher detection accuracy, its network computational complexity is greater, resulting in significantly slower detection speed. Our method increased the mAP by 4.55% and improved the FPS by 25. Compared to the improved YOLOv4-tiny algorithm proposed by Sharma et al. [44], while this method achieves an excellent FPS of 129, its computational load is too high, and the mAP is only 64.90, which is 16.13% lower than our YOLOv7-tiny-RCA method. These results demonstrate that our method reduces the number of model parameters and computational load while achieving high-precision traffic sign detection.

The YOLOv7-tiny-RCA model achieves an mAP of 81.03%, significantly outperforming YOLOv4-tiny and YOLOv5s. This improvement is largely attributed to the integration of the ELAN-REP module and the CBAM attention mechanism, which enhance the model's ability to precisely localize and detect traffic signs even from a distance. The ELAN-REP module reduces redundancy in the model's architecture, contributing to a more efficient computation as evidenced by a reduction in GFLOPS and Params. Specifically, the model sees a reduction in parameters from 6.13M in the base YOLOv7-tiny model to 4.99M in the RCA version, highlighting a more streamlined and efficient network.

FPS improvements are particularly notable in scenarios demanding real-time processing. Our model maintains high FPS rates conducive to deployment in dynamic environments such as moving vehicles. The AFPN feature fusion network further aids this by optimizing the interaction between layers, allowing for faster feature processing and improved real-time performance metrics.

Tables and Figures illustrating where YOLOv7-tiny-RCA provides the best improvements in detection accuracy compared to its predecessors. For example, in tests involving occluded or distant signs, the AFPN's ability to integrate multi-scale features has shown to be especially beneficial, resulting in higher precision and recall rates.

The implementation of the SIOU loss function enhances the model's convergence rate and robustness. Unlike traditional IoU-based loss functions, SIOU accounts for the orientation and aspect ratio of the traffic signs, which is critical for accurate detection in cluttered street scenes.

To validate the advancement of the YOLOv7-tiny-RCA algorithm in the field of traffic sign detection, we extended our comparison to include other traffic sign detection methods on the TT100K-B dataset. Table 4 presents the comparative experimental results of the YOLOv7-tiny-RCA method against other traffic sign detection methods on the TT100K-B dataset. As demonstrated by the experimental data in Table 4,

TABLE 5. Comparison of detection results on the TT100K-B and CCTSDB datasets.

Model	Data	mAP@0.5%	FPS	Pramas(M)	GFLOPS
YOLOv7-tiny	TT100K-B	77.12	110	6.13	13.6
YOLOv7-tiny-RCA（Ours）	**TT100K-B**	**81.03**	**104**	**4.99**	**13.3**
YOLOv7-tiny	CCTSDB	84.68	107	6.13	14.0
YOLOv7-tiny-RCA（Ours）	**CCTSDB**	**87.91**	**102**	**4.99**	**13.6**

the YOLOv7-tiny-RCA method proposed in this paper shows a significant improvement in mAP compared to the traffic sign detection method improved from YOLOv7-tiny by Jia et al. Compared to the baseline model YOLOv7-tiny, the mAP improved by 9.72%. Although the TSDet method has a higher detection accuracy, its network computational complexity is greater, resulting in significantly slower detection speed. Our method increased the mAP by 4.55% and improved the FPS by 25. Compared to the improved YOLOv4-tiny algorithm proposed by Sharma et al., while this method achieves an excellent FPS of 129, its computational load is too high, and the mAP is only 64.90, which is 16.13% lower than our YOLOv7-tiny-RCA method. These results demonstrate that our method reduces the number of model parameters and computational load while achieving high-precision traffic sign detection.

Additionally, we conducted further experiments using the CCTSDB dataset to validate the generalization capability of YOLOv7-tiny-RCA. The experimental results in Table 5 show that the improved YOLOv7-tiny-RCA model (Ours) significantly outperforms the baseline model YOLOv7-tiny in detection performance on both the TT100K-B and CCTSDB datasets. On the CCTSDB dataset, the mAP@0.5% of YOLOv7-tiny-RCA increased from 84.68% (baseline YOLOv7-tiny) to 87.91%. Although the FPS decreased slightly from 107 to 102, the model's parameter count and computational load were reduced to 4.99M and 13.6 GFLOPS, respectively. Overall, these results indicate that the optimized YOLOv7-tiny-RCA model demonstrates excellent detection performance on both the TT100K-B and CCTSDB datasets. It significantly improves detection accuracy while maintaining high real-time performance and low computational complexity, fully validating the effectiveness, robustness, and generalization capability of our method across different datasets.

F. VISUALIZATION

To more intuitively demonstrate the performance advantages of the proposed improved algorithm YOLOv7-tiny-RCA, this study selected scenarios with distant traffic signs and deformed shapes for testing YOLOv7-tiny-RCA against YOLOv4-tiny, YOLOv5s, YOLOv7, and the baseline model YOLOv7-tiny. The visualized test results are shown in Figure 15 and Figure 16, respectively.

Figure 15 shows the detection results of different models in distant scenarios. As illustrated, YOLOv7-tiny-RCA

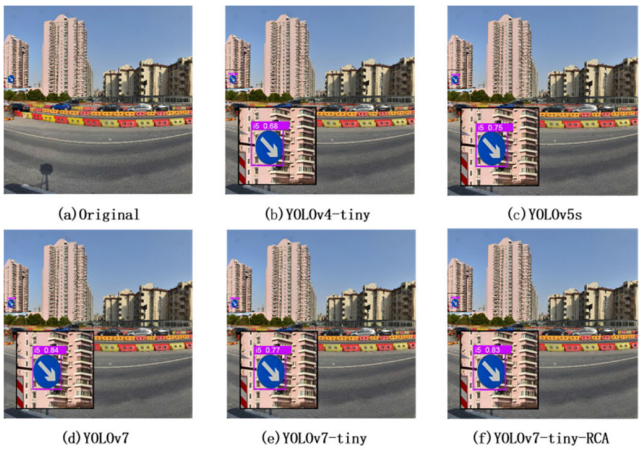


FIGURE 15. Comparison of detection results for distant traffic signs across different models.

demonstrates a significant performance advantage in detecting distant i5 traffic signs. Compared to YOLOv7, which has a larger parameter count and greater computational complexity, the confidence score of YOLOv7-tiny-RCA is only 0.01 lower. Specifically, YOLOv7 has a confidence score of 0.84, while YOLOv7-tiny-RCA achieves 0.83, indicating a substantial reduction in computational resource consumption with minimal loss in accuracy. Additionally, the detection results of YOLOv7-tiny-RCA show improvements of 0.15 and 0.08 over the YOLOv4-tiny and YOLOv5s models, respectively, demonstrating a clear advantage in the detection accuracy of distant signs.

As observed in Figure 16, YOLOv4-tiny experiences missed detections when dealing with traffic signs that have undergone shape distortion. In contrast, the YOLOv7-tiny-RCA algorithm not only accurately detects the pl40 category of signs but also achieves a detection score of 0.79, significantly higher than the 0.71 and 0.72 achieved by YOLOv5s and YOLOv7-tiny, respectively. This demonstrates the adaptability and robustness of YOLOv7-tiny-RCA in handling deformed traffic signs.

The YOLOv7-tiny-RCA algorithm demonstrates exceptional performance in distant and complex scenarios. While maintaining a lightweight design, the algorithm has improved detection accuracy and reliability through optimization. It is suitable for deployment in resource-constrained edge devices, effectively enhancing the efficiency and accuracy of real-time monitoring. In summary, YOLOv7-tiny-RCA reduces

FIGURE 16. Comparison of detection results for deformed traffic signs across different models.

computational costs while successfully enhancing the capability to detect distant and deformed traffic signs through meticulous adjustment and optimization of the algorithm. Its outstanding performance indicates that the algorithm can meet the demands for efficient and accurate traffic sign detection in today's intelligent transportation systems.

V. CONCLUSION

This study proposes the YOLOv7-tiny-RCA, a traffic sign recognition method that achieves a balanced approach between lightweight design and real-time performance. Initially, an efficient layer aggregation network module, ELAN-REP, was developed using RepConv to replace the original ELAN module in the backbone network. Subsequently, the CBAM was integrated during the backbone feature extraction stage to enhance the capture of positional information. Additionally, a new feature fusion method, AFPN, was introduced at the network neck to ensure the integrity of information transfer across multiple levels while reducing the model's parameter count. Finally, the CIOU loss used by YOLOv7-tiny was replaced with SIOU to further improve the model's convergence speed, accuracy, and regression rate. Moreover, a series of ablation studies, comparative experiments, and visual analyses were conducted on the class-balanced, data-augmented traffic sign dataset TT100K-B for YOLOv7-tiny-RCA. Experimental results indicate that the proposed YOLOv7-tiny-RCA achieved an mAP of 81.03%, optimized the parameter count to 4.99M, and maintained GFLOPS at 13.3G during the training phase, accomplishing a reduction in model computational load and parameter count alongside an enhancement in accuracy. Additionally, to validate the generalizability of the improved method, experiments were also conducted on the CCTSDB dataset, achieving an mAP of 87.91%.

While the proposed YOLOv7-tiny-RCA model enhanced with CBAM and AFPN demonstrates improved traffic sign detection, particularly on edge devices, it faces potential limitations in generalization across varied environmental conditions and detecting occluded or partially visible signs. To address these, enhancing dataset diversity through synthetic data and employing domain adaptation could improve model robustness across different scenarios. Additionally, integrating sophisticated spatial attention mechanisms or generative models to handle occlusions could further refine the detection accuracy. Balancing these enhancements with the computational efficiency required for real-time processing on edge devices remains a critical challenge, necessitating further optimization strategies to maintain performance without compromising speed.

REFERENCES

[1] 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, Jun. 2017.
[2] J. Li and Z. Wang, "Real-time traffic sign recognition based on efficient CNNs in the wild," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 3, pp. 975–984, Mar. 2019.
[3] D. Tabernik and D. Skocaj, "Deep learning for large-scale traffic-sign detection and recognition," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1427–1440, Apr. 2020.
[4] A. Pon, O. Adrienko, A. Harakeh, and S. L. Waslander, "A hierarchical deep architecture and mini-batch selection method for joint traffic sign and light detection," in *Proc. 15th Conf. Comput. Robot Vis. (CRV)*, May 2018, pp. 102–109.
[5] J. Zhang, M. Huang, X. Jin, and X. Li, "A real-time Chinese traffic sign detection algorithm based on modified YOLOv2," *Algorithms*, vol. 10, no. 4, p. 127, Nov. 2017.
[6] S. P. Rajendran, L. Shine, R. Pradeep, and S. Vijayaraghavan, "Real-time traffic sign recognition using YOLOv3 based detector," in *Proc. 10th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT)*, Jul. 2019, pp. 1–7.
[7] J. Chen, K. Jia, W. Chen, Z. Lv, and R. Zhang, "A real-time and high-precision method for small traffic-signs recognition," *Neural Comput. Appl.*, vol. 34, no. 3, pp. 2233–2245, Feb. 2022.
[8] X. Luo, Z. Li, and H. Zhu, "Traffic sign detection based on YOLOv4-ghost," *Chin. Comput. Digit. Eng.*, vol. 50, no. 6, pp. 1292–1297, 2022.
[9] X. Yang, W. Jiang, and H. Yuan, "Traffic sign recognition detection based on YOLOv5," *Inf. Technol. Informatization*, vol. 46, no. 4, pp. 28–30, 2021.
[10] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, "Traffic-sign detection and classification in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2110–2118.
[11] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.
[12] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. 14th Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands. Cham, Switzerland: Springer, Oct. 2016, pp. 21–37.
[13] T. Liang, H. Bao, W. Pan, and F. Pan, "Traffic sign detection via improved sparse R-CNN for autonomous vehicles," *J. Adv. Transp.*, vol. 2022, pp. 1–16, Mar. 2022.
[14] J. Wang, Y. Chen, X. Ji, Z. Dong, M. Gao, and C. S. Lai, "Metaverse meets intelligent transportation system: An efficient and instructional visual perception framework," *IEEE Trans. Intell. Transp. Syst.*, early access, May 22, 2024, doi: 10.1109/TITS.2024.3398586.
[15] J. Wang, Y. Chen, Y. Gu, Y. Yan, Q. Li, M. Gao, and Z. Dong, "A lightweight vehicle mounted multi-scale traffic sign detector using attention fusion pyramid," *J. Supercomput.*, vol. 80, no. 3, pp. 3360–3381, Feb. 2024.
[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 779–788.
[17] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 7263–7271.
[18] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
[19] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.

[20] C. Zhang, X. Hu, and H. Niu, "Vehicle object detection based on improved YOLOv5 method," *J. Sichuan Univ., Natural Sci. Ed.*, vol. 59, no. 5, pp. 73–81, 2022.
[21] S. Du, W. Pan, N. Li, S. Dai, B. Xu, H. Liu, C. Xu, and X. Li, "TSD-YOLO: Small traffic sign detection based on improved YOLO v8," *IET Image Process.*, Jun. 2024, doi: 10.1049/ipr2.13141.
[22] C.-Y. Wang, A. Bochkovskiy, and H.-Y.-M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 7464–7475.
[23] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size," 2016, *arXiv:1602.07360*.
[24] A. Gholami, K. Kwon, B. Wu, Z. Tai, X. Yue, P. Jin, S. Zhao, and K. Keutzer, "SqueezeNext: Hardware-aware neural network design," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 171900–171909.
[25] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6848–6856.
[26] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "GhostNet: More features from cheap operations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1577–1586.
[27] T. Liang, H. Bao, W. Pan, and F. Pan, "ALODAD: An anchor-free lightweight object detector for autonomous driving," *IEEE Access*, vol. 10, pp. 40701–40714, 2022.
[28] Y. Tang, K. Han, J. Guo, C. Xu, C. Xu, and Y. Wang, "GhostNetv2: Enhance cheap operation with long-range attention," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 9969–9982.
[29] A. Howard, M. Sandler, B. Chen, W. Wang, L.-C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le, "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1314–1324.
[30] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.
[31] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.
[32] Y. Lee, J.-W. Hwang, S. Lee, Y. Bae, and J. Park, "An energy and GPU-computation efficient backbone network for real-time object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 752–760.
[33] C.-Y. Wang, H.-Y. Mark Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 1571–1580.
[34] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944.
[35] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8759–8768.
[36] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, "RepVGG: Making VGG-style ConvNets great again," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13728–13737.
[37] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
[38] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.
[39] G. Yang, J. Lei, Z. Zhu, S. Cheng, Z. Feng, and R. Liang, "AFPN: Asymptotic feature pyramid network for object detection," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2023, pp. 2184–2189.
[40] S. Liu, D. Huang, and Y. Wang, "Learning spatial fusion for single-shot object detection," 2019, *arXiv:1911.09516*.
[41] Z. Gevorgyan, "SIoU loss: More powerful learning for bounding box regression," 2022, *arXiv:2205.12740*.
[42] Z. Jia, S. Sun, and G. Liu, "Real-time traffic sign detection based on weighted attention and model refinement," *Neural Process. Lett.*, vol. 55, no. 6, pp. 7511–7527, Dec. 2023.
[43] Y. J. Qian and B. Wang, "TSDet: A new method for traffic sign detection based on YOLOv5-SwinT," *IET Image Process.*, vol. 18, no. 4, pp. 875–885, Mar. 2024.
[44] V. K. Sharma, P. Dhiman, and R. K. Rout, "Improved traffic sign recognition algorithm based on YOLOv4-tiny," *J. Vis. Commun. Image Represent.*, vol. 91, Mar. 2023, Art. no. 103774.

XIAOBING CAO received the B.S. and Ph.D. degrees from Nanjing University of Science and Technology, Nanjing, China, in 2003 and 2012, respectively. He is currently an Associate Professor with the School of Control Engineering, Wuxi Institute of Technology. His current research interests include machine vision and mobile robots.

YICEN XU received the M.S. degree from Nanjing University of Science and Technology, in 2006. She is currently an Associate Professor with the School of Intelligent Equipment and Automotive Engineering, Wuxi Vocational Institute of Commerce. Her current research interests include intelligent robots and intelligent manufacturing.

JIAWEI HE received the M.S. degree in mechanical engineering from Jiangnan University, Wuxi, China, in 2016. She is currently an Engineer with Wuxi Institute of Technology. Her research interests include machine vision, image processing, and the integration technology of automation systems.

JIAHUI LIU received the B.S. degree in engineering from Mudanjiang Normal University, Mudanjiang, China, in 2020. She is currently pursuing the M.S. degree in computer technology with Northeast Forestry University, Harbin, China. She is a Researcher with the Applied Software Research Institute, Northeast Forestry University. Her research interests include object detection and image classification.

YONGJIE WANG received the B.S. degree in metal materials engineering from Yantai University, Yantai, China, in 2021. He is currently pursuing the M.S. degree in computer technology with Northeast Forestry University, Harbin, China. He is a Researcher with the Applied Software Research Institute, Northeast Forestry University. His research interests include object detection, image classification, and person re-identification.