Journal of Image Processing Theory and Applications (2026)
Clausius Scientific Press, Canada

DOI: 10.23977/jipta.2026.090105
ISSN 2560-6239 Vol. 9 Num. 1

A Multi-Class Traffic Sign Detection Algorithm Based on an Improved YOLOv13

Guolin Cong[1,a], Su Xu[1,b,*], Chong Zheng[1,c]

[1]*School of Electronic Engineering, Jiangsu Ocean University, Lianyungang, Jiangsu, 222000, China*
[a]*2479827904@qq.com*, [b]*24354293@qq.com*, [c]*1719275326@qq.com*
[*]*Corresponding author*

Keywords: Traffic Sign Detection; YOLOv13n; Lightweight; ShuffleNetV2; GSConv; MBConv; Inner-CIoU

Abstract: To address the problems of missed detection of long-distance small traffic signs, insufficient robustness in complex road environments, and limited deployment on vehicle-mounted embedded platforms, a lightweight traffic sign detection algorithm, termed TS-YOLOv13, is proposed based on YOLOv13n. First, the original backbone network is replaced with ShuffleNetV2 to reduce model parameters and memory access overhead while enhancing shallow feature extraction. Second, a lightweight Slim Neck based on GSConv is introduced to improve the efficiency of multi-scale feature fusion. Third, the detection head is reconstructed using the Mobile Inverted Bottleneck Convolution (MBConv), which further reduces computational cost while maintaining detection performance. Finally, the Inner-CIoU loss function is employed to optimize bounding box regression, thereby improving the localization accuracy of hard samples and small traffic signs. To evaluate the effectiveness of the proposed method, ablation and comparative experiments are conducted on the TT100K dataset, while its generalization capability is further validated on the CCTSDB, GTSDB, and LISA datasets. Experimental results demonstrate that, compared with the original YOLOv13n, TS-YOLOv13 achieves a precision of 70.5%, a recall of 57.0%, and an mAP@0.5 of 63.8%. Meanwhile, the number of parameters is reduced from 2.45M to 2.07M, the computational complexity is decreased from 6.2GFLOPs to 5.1GFLOPs, and the model size is reduced from 5.4MB to 4.6MB. Furthermore, compared with several mainstream lightweight object detection algorithms, TS-YOLOv13 achieves a better trade-off among detection accuracy, model size, and inference efficiency, while maintaining good generalization performance across multiple public datasets. The proposed algorithm improves detection accuracy while achieving model lightweighting, making it suitable for real-time traffic sign detection on vehicle-mounted embedded platforms.

1. Introduction

With the rapid development of Intelligent Transportation Systems (ITS) and autonomous driving technologies, traffic signs have become important carriers of road traffic information, providing critical information such as speed limits, prohibitions, warnings, and directions. The accuracy of

traffic sign detection directly affects vehicle perception, path planning, and decision-making. However, in real road environments, traffic signs are often characterized by small sizes at long distances, large scale variations, frequent occlusions, complex backgrounds, and changing illumination conditions, which increase the difficulty of accurate detection. Meanwhile, vehicle-mounted embedded platforms are limited by computational resources, storage capacity, and power consumption, placing higher requirements on the accuracy, real-time performance, and lightweight design of traffic sign detection algorithms. Therefore, developing a traffic sign detection algorithm with high accuracy, low computational cost, and good deployment performance is of great theoretical significance and practical value.

Traffic sign detection has mainly evolved through two stages: traditional computer vision methods and deep learning-based methods. Traditional detection methods mainly rely on color segmentation, edge detection[1], template matching, and handcrafted features such as the Histogram of Oriented Gradients (HOG), combined with classifiers for target detection. However, these methods have limited adaptability to complex backgrounds and illumination variations, resulting in poor robustness. With the rapid development of deep learning, convolutional neural network (CNN)-based object detection algorithms have gradually become the mainstream. Two-stage detectors represented by Faster R-CNN[2] achieve high detection accuracy but suffer from complex inference procedures, making them unsuitable for real-time applications. In contrast, one-stage detectors represented by the YOLO series[3] perform end-to-end object detection with a good balance between accuracy and speed, and have become the mainstream framework for traffic sign detection.

In recent years, extensive research has been conducted to address the problems of missed detection of small traffic signs, insufficient adaptability to complex road scenes, and the difficulty of deploying detection models on edge devices. For multi-scale feature enhancement, Han et al.[4] proposed a Hierarchical Multi-Scale Fusion Network (HMSF-Net) combined with a Multi-Scale Swin Transformer (MS-Swin) to improve multi-scale feature representation, thereby enhancing the detection accuracy of traffic signs in complex scenes. However, the introduction of the Transformer architecture significantly increased the computational cost of the model. Ma et al.[5] proposed the SPPCAKO feature extraction module and the SDI-Damo Neck adaptive feature fusion structure, which enhanced feature extraction and multi-scale information fusion, leading to improved detection accuracy in complex environments. Nevertheless, the network became more complicated and required higher training costs. Dong et al.[6] designed a TriplePathBlock together with a dynamic fusion gate mechanism to strengthen multi-path feature representation and improve the robustness of traffic sign detection. However, the parallel attention structure increased inference latency and limited real-time performance.

In terms of lightweight network design, Xu et al.[7] proposed the YOLO-SAL model by introducing the SCC2f lightweight convolution module and Long-Sequence Knowledge Attention (LSKA), which improved detection performance while reducing the number of parameters and computational cost. However, its high-level semantic representation remained insufficient in complex scenes, resulting in unstable performance under occlusion and backlighting conditions. Guo et al.[8] proposed RBL-YOLOv8, which optimized the backbone using the RepNCSPELAN module and introduced weighted cross-layer feature fusion to improve small-object detection. However, missed detections still occurred in heavily occluded scenes.

For small-object detection, Zhang et al.[9] developed a Multi-layer Interactive Residual Feature Fusion Network (MIRFFN) and introduced Stage Routing Attention (SRA) to strengthen feature interaction between shallow and deep layers, thereby improving the detection accuracy of small targets. However, the cross-layer fusion structure increased model complexity and reduced inference efficiency. Zhang et al.[10] designed a Region Feature Enhancement Module (RFEM) and added an extra detection layer for small objects, which effectively improved the detection accuracy of distant

traffic signs. Nevertheless, the overall network architecture was not redesigned for lightweight deployment, resulting in relatively high deployment costs. Han et al.[11] introduced Space-to-Depth Convolution (SPD-Conv) and a Context Enhancement Module (CAM) to improve feature extraction for low-resolution small targets. However, the loss function was not optimized for difficult samples, leaving room for improvement in the localization accuracy of small and occluded traffic signs.

Overall, existing studies mainly focus on optimizing a single aspect of the detection framework. Some methods emphasize detection accuracy while neglecting lightweight deployment, whereas others prioritize model compression at the expense of performance in complex scenarios. Consequently, it remains challenging to achieve a good balance among detection accuracy, small-object detection performance, and real-time inference efficiency for vehicle-mounted embedded applications.

To address the above issues, this paper takes YOLOv13n as the baseline model and proposes a lightweight optimization method for real-time traffic sign detection in vehicle-mounted scenarios. By improving the network architecture, feature fusion strategy, and detection optimization method, the proposed approach reduces the computational complexity and model size while enhancing the feature representation and localization accuracy of multi-scale traffic signs. As a result, it achieves a better balance among detection accuracy, model lightweightness, and real-time inference performance, providing an effective solution for deploying traffic sign detection algorithms on vehicle-mounted embedded platforms.

2. YOLOv13 Algorithm

YOLOv13 is a new-generation one-stage object detection algorithm developed through iterative improvements based on YOLOv11 and YOLOv12. It introduces architectural innovations to address the limitations of previous models, including the restricted receptive field of convolutional operations, insufficient global feature modeling capability, and the high computational cost of self-attention mechanisms. As a result, YOLOv13 integrates object detection with high-level semantic feature modeling to achieve improved detection performance. Following the scaling strategy of the YOLO series, YOLOv13 provides five model variants ranging from YOLOv13-n to YOLOv13-x. The model depth and channel width are gradually increased by adjusting the scaling coefficients, allowing users to customize the model according to different training and inference requirements. In this paper, the lightweight YOLOv13n model is selected as the baseline because it is well suited to the computational constraints of vehicle-mounted embedded platforms.

YOLOv13 exhibits excellent cross-platform compatibility and can be efficiently deployed on various hardware platforms, including CPUs, GPUs, FPGAs, and vehicle-mounted edge computing chips, without requiring complex software configurations. Therefore, it is suitable for a wide range of application scenarios. In intelligent driving applications, the limited computing capability and power constraints of vehicle-mounted embedded platforms impose strict requirements on the inference efficiency and energy consumption of detection algorithms. Benefiting from its lightweight design based on depthwise separable convolution, together with the HyperACE global feature modeling module and the FullPAD full-path feature distribution mechanism, YOLOv13 achieves a good balance between detection accuracy and computational efficiency, making it well suited for real-time traffic sign detection on vehicle-mounted platforms. Figure 1 illustrates the overall architecture of YOLOv13, showing the connections and functions of its major network components.

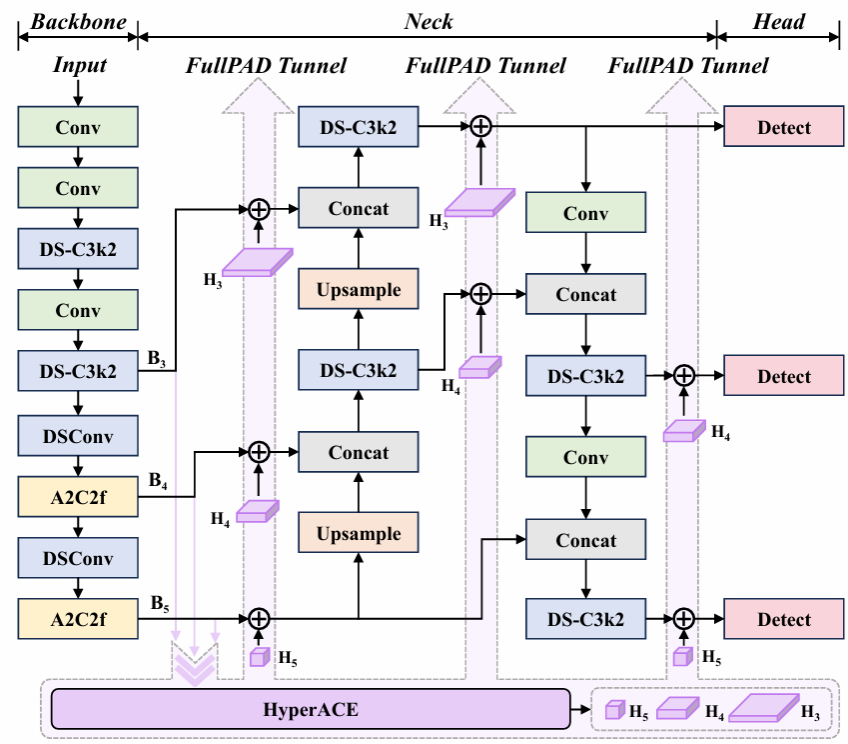

Figure 1: Overall Architecture of YOLOv13.

2.1. Main Structure of YOLOv13

The overall structure of YOLOv13 consists of three parts: Backbone, Neck, and Head.

The Backbone mainly includes core modules such as DSConv and DS-C3k2, which construct a lightweight feature extraction network based on depthwise separable convolution. Among them, DS-C3k2 is the core unit of the Backbone. It integrates depthwise separable convolution into the traditional CSP bottleneck structure, reducing redundant parameters through lightweight design while maintaining feature extraction capability and computational efficiency.

The Neck adopts the HyperACE adaptive hypergraph association module and the FullPAD full-path feature distribution mechanism to fuse feature maps from different levels of the Backbone. It achieves high-order correlation integration of multi-scale features and improves the global feature representation capability of the model.

The Head adopts a decoupled structure, consisting of independent regression and classification branches. The regression branch performs object localization, while the classification branch is responsible for category recognition.

2.2. Core Modules of YOLOv13

DSConv: DSConv, namely depthwise separable convolution, is the core basic unit of lightweight networks. Unlike standard convolution, which simultaneously performs spatial feature extraction and channel dimension fusion, DSConv divides the computational process into two stages: first, depthwise convolution (DWConv) independently performs k×k convolution operations on each input channel to extract spatial features within a single channel; then, pointwise convolution (PWConv) uses a 1×1 convolution kernel to perform linear mapping in the channel dimension, achieving cross-channel information fusion and channel dimension adjustment. This structure can complete equivalent feature transformations while significantly reducing the number of model parameters and

floating-point computation. In the engineering implementation of YOLOv13, the output of pointwise convolution is sent to the batch normalization (BN) layer and then processed by the SiLU activation function. Together with DWConv and PWConv, it forms a complete feature transformation unit. This module is mainly deployed in the Backbone network, performing lightweight feature extraction and downsampling functions. Its overall structure is shown in Figure 2.

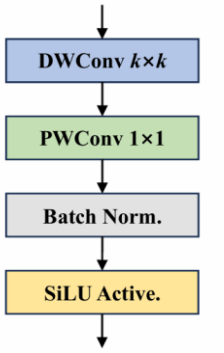

Figure 2: Structure of DSConv.

DS-C3k2: DS-C3k2 first uses 1×1 pointwise convolution to adjust the input channel dimensions. After feature dimension reduction, the features are divided into two parallel branches through channel splitting: one is the shortcut branch, which directly transfers features to the downstream and preserves the original feature information; the other is the main feature branch, which stacks multiple DS-C3k lightweight bottleneck units for deep feature enhancement. Each DS-C3k unit adopts depthwise separable convolution to replace standard convolution, significantly reducing the number of parameters and computational cost while maintaining feature extraction capability. After the two branches complete feature processing, the features are concatenated and fused through channel concatenation, and finally a 1×1 pointwise convolution is used to integrate channel information and output the final features. This module adopts the parallel branch design idea, dividing the traditional cascaded bottleneck structure into dual-channel parallel operations. By replacing the original standard convolution with depthwise separable convolution, it reduces redundant computation and compresses the number of parameters while enriching gradient propagation paths, effectively alleviating the gradient vanishing problem in deep networks. Meanwhile, the number of internal DS-C3k stacking units can be flexibly adjusted according to task requirements. The structure of DS-C3k2 is shown in Figure 3.

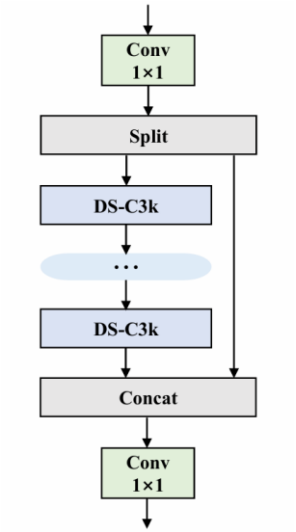

Figure 3: Structure of DS-C3k2.

HyperACE: HyperACE is a representative innovation module of YOLOv13. To address the limited receptive field of conventional convolution and the high computational cost of traditional self-attention, it adopts hypergraph convolution and high-low order branch design to achieve efficient multi-scale high-order feature correlation modeling. Its structure is shown in Figure 4. The module takes three feature maps with different resolutions, B_3, B_4, and B_5, output from the Backbone network as inputs. First, the shallow high-resolution feature B_3 is downsampled, and the deep low-resolution feature B_5 is upsampled to unify their scales with the resolution of the intermediate feature B_4. The three feature maps are concatenated along the channel dimension and then adjusted by 1×1 pointwise convolution to complete the initial integration of multi-scale features. Subsequently, the features are divided into two parallel branches through channel splitting. The high-order branch stacks multiple C3AH modules and relies on adaptive hyperedge generation and hypergraph convolution to achieve pixel-level many-to-many correlation modeling. It dynamically learns the participation weights of each feature point in hyperedges and adaptively generates hypergraph topology structures, capturing global semantic correlations between targets across different scales and positions, and compensating for the limitation of conventional convolution with restricted receptive fields. The low-order branch adopts the DS-C3k lightweight convolution unit, focusing on local spatial features and detailed texture extraction. It preserves low-level feature information with extremely low computational cost and forms complementary characteristics with the high-order branch while controlling the overall computational complexity of the module. After the two branches complete feature processing, the features are fused through channel concatenation, and finally a 1×1 pointwise convolution is used to integrate channel information and output enhanced multi-scale features. Combined with the FullPAD full-path feature distribution mechanism, the enhanced features can be transmitted back to different levels of the network layer by layer, improving the information flow efficiency of the entire network and enhancing the overall feature representation capability of the model.

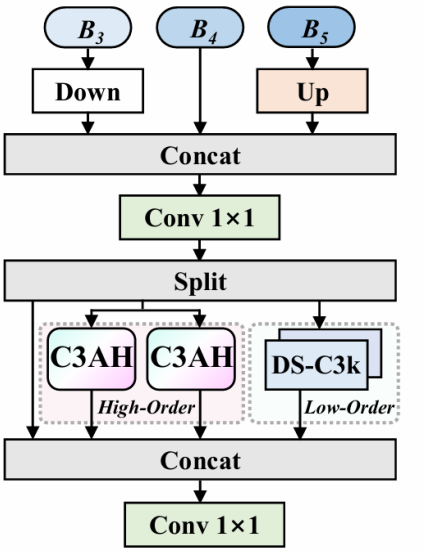

Figure 4: Structure of HyperACE.

Detection Head Module: The Detection Head of YOLOv13 adopts a decoupled structure, consisting of a regression branch and a classification branch. The regression branch uses two stacked standard convolution layers, followed by the DFL distribution integral module to output refined bounding box coordinates. The classification branch adopts two stacked depthwise separable convolution layers to output the probability distribution of each category. The two branches are trained independently and work cooperatively to complete the detection task. The structure of the Detection Head is shown in Figure 5.

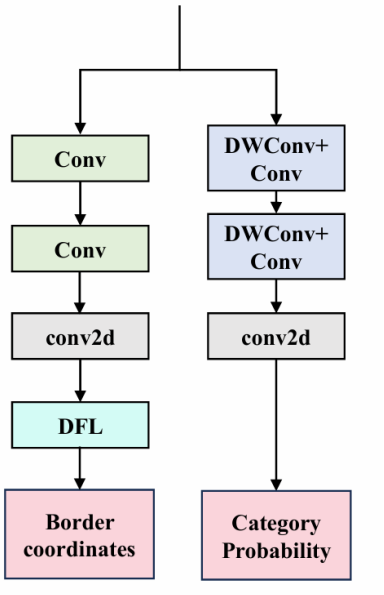

Figure 5: Structure of Detection Head.

3. Improved TS-YOLOv13 Algorithm

For multi-class traffic sign detection tasks in vehicle-mounted embedded scenarios, considering the limited computing resources and power consumption of terminal devices, as well as the challenges of a high proportion of small targets, frequent occlusion, and significant scale variations in real-world images, this paper selects YOLOv13n as the baseline network for targeted optimization. Although the original YOLOv13n has a lightweight foundation based on depthwise separable convolution, it still has four limitations: first, the memory access cost of the Backbone is relatively high, and shallow detail information of small targets is easily lost during multiple downsampling processes; second, the multi-scale feature interaction capability of the Neck fusion module is limited, affecting feature fusion performance in complex scenarios; third, the Detection Head uses stacked standard convolutions, resulting in parameter and computational redundancy and increasing the deployment burden on embedded platforms; fourth, the default CIoU loss function adopts a unified optimization strategy for samples with different difficulties, resulting in insufficient bounding box regression accuracy for difficult samples.

To address the above problems, this paper conducts a systematic improvement from four aspects: feature extraction, feature fusion, detection output, and loss optimization, aiming at lightweight deployment and improved detection accuracy. ShuffleNetV2 is used to replace the original Backbone network, reducing model size while enhancing the preservation ability of shallow features for small targets; a Slim Neck lightweight Neck structure is constructed based on GSConv to improve multi-scale feature fusion performance; the MBConv module is introduced to reconstruct the Detection Head, reducing computational and storage overhead during detection; the Inner-CIoU loss function is adopted to optimize the bounding box regression process and improve the localization accuracy of difficult samples. The improved algorithm is named TS-YOLOv13, and its structure is shown in Figure 6.

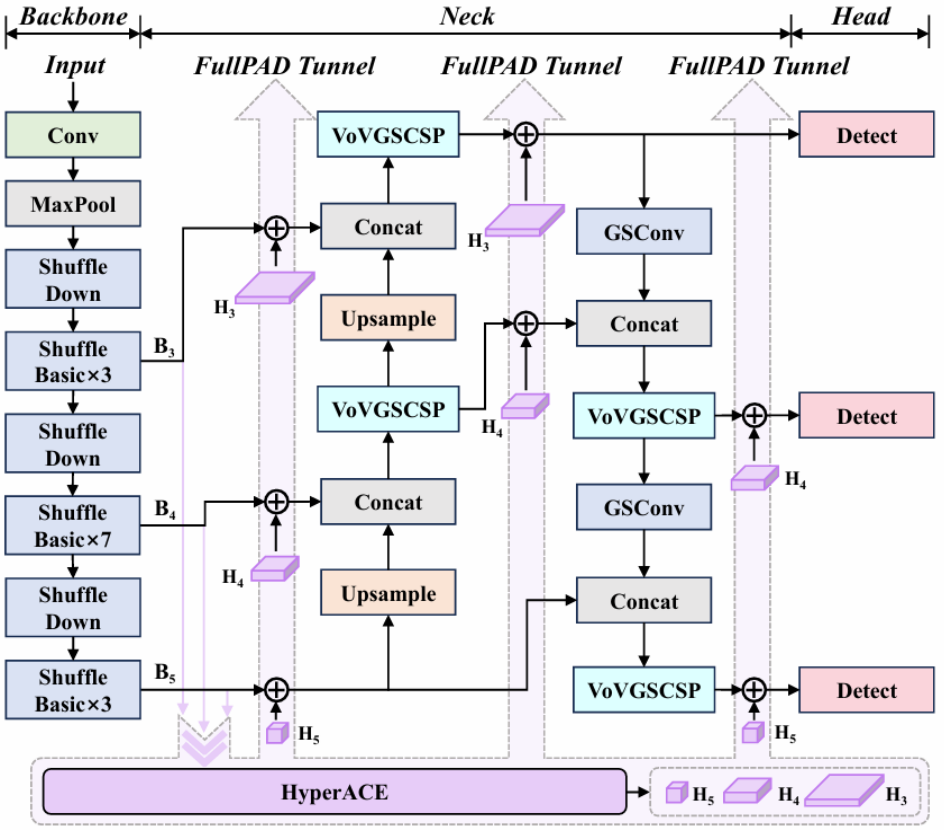

Figure 6: Overall Architecture of TS-YOLOv13.

3.1. Lightweight Backbone Network Based on ShuffleNetV2

The original Backbone network of YOLOv13n is constructed by stacking DSConv depthwise separable convolution and DS-C3k2 modules. Although it provides a lightweight foundation, the overall network has relatively high memory access cost, and shallow texture features of distant small traffic signs are easily lost during multiple downsampling processes. On vehicle-mounted embedded devices with limited computing resources and memory bandwidth, the actual inference efficiency still has room for improvement. To reduce model storage consumption, optimize memory access efficiency, and enhance the feature retention capability of small targets, this paper replaces the original Backbone network of YOLOv13n with ShuffleNetV2.

ShuffleNetV2 takes minimizing memory access cost and maximizing parallelism as its core design principles. It proposes four efficient network design principles for mobile devices and specifically addresses the high memory consumption caused by a large number of group convolutions in early lightweight networks[12]. Its core innovations are the Channel Split and Channel Shuffle mechanisms, and it contains two types of basic building units, as shown in Figure 7. The basic unit has a stride of 1 and does not change the size of the feature map. Its operation process is as follows: the input feature is split into two parts along the channel dimension. One branch is directly connected through a shortcut connection to achieve feature reuse without additional convolution computation. The other branch removes grouped 1×1 convolution and sequentially performs 1×1 pointwise convolution, 3×3 depthwise convolution, and 1×1 pointwise convolution, reducing the channel isolation effect caused by group operations. After the two branches are concatenated, the Channel Shuffle operation is performed to disrupt the channel arrangement and achieve information interaction between different branches. The downsampling unit has a stride of 2 and is used to compress the feature resolution.

This unit removes the channel split operation and divides the input feature into two independent convolution branches. Both branches use depthwise convolution with a stride of 2 for downsampling, and the number of output channels is doubled, expanding the feature dimension while compressing the feature scale.

The overall ShuffleNetV2 Backbone network consists of an initial convolution layer, a maximum pooling layer, and three-stage stacked units. The first layer of each Stage contains one downsampling unit for resolution reduction, and several basic units are stacked internally for deep feature extraction. The numbers of repeated basic units in the three Stages are 3, 7, and 3, respectively, generating multi-scale feature maps with 8×, 16×, and 32× downsampling rates. After replacing the Backbone, the model parameters and storage size are effectively reduced, and the memory access efficiency is significantly improved. Meanwhile, the shortcut branch can preserve complete shallow texture information, and the Channel Shuffle operation enhances cross-channel feature fusion. Therefore, the proposed method can better retain the detailed features of distant small traffic signs while achieving a balance between vehicle-mounted deployment adaptability and small target feature extraction capability.

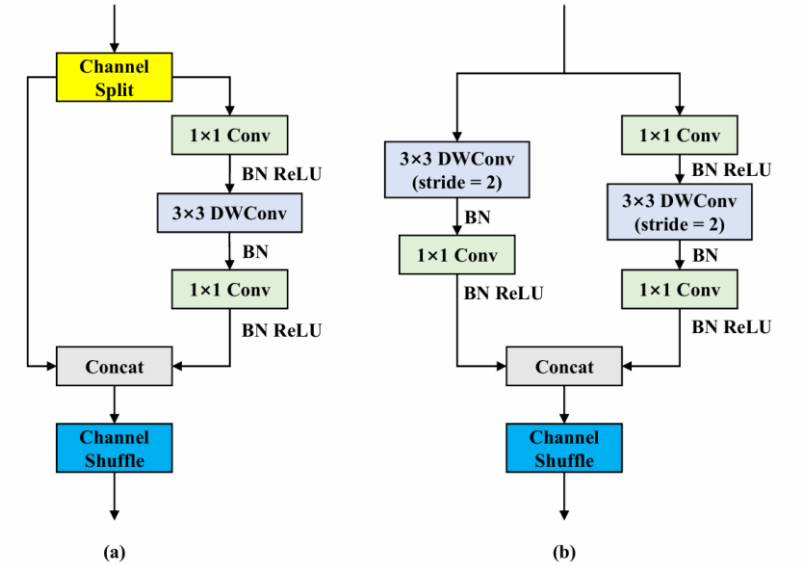

Figure 7: Structure of ShuffleNetV2 Units. (a) Basic Unit; (b) Downsampling Unit.

3.2. Slim Neck Network Based on GSConv

The original Neck network of YOLOv13n uses DS-C3k2 and standard convolution to perform downsampling and multi-scale feature concatenation and fusion. The lightweight unit DS-C3k2 in the original network relies on pure depthwise separable convolution for feature extraction, which has the problem of channel separation. The lack of information interaction between channels leads to poor feature integrity of occluded and deformed traffic signs after fusion. Meanwhile, the Neck downsampling process uses standard convolution, resulting in relatively high computational cost. To address the above problems, this paper constructs a Slim Neck network based on GSConv[13] to achieve lightweight reconstruction of the Neck structure.

GSConv adopts a hybrid design of standard pointwise convolution and depthwise separable convolution, and its structure is shown in Figure 8. The input feature is first reduced in dimension through 1×1 standard pointwise convolution to generate channel-dense features, and then divided into two branches: one branch directly retains the original pointwise convolution results, while the other branch uses 3×3 depthwise convolution to extract spatial features. The two branch features are

concatenated along the channel dimension and then processed by channel shuffle operation, allowing dense channel information to evenly penetrate into the depthwise convolution branch and compensating for the limitation of isolated channel information in pure depthwise separable convolution. Based on GSConv, the GS bottleneck residual module is constructed and further embedded into the CSP split structure to form the VoVGSCSP module. After feature input, it is divided into two branches: one branch extracts deep features through multiple stacked GS bottleneck modules, while the other branch performs shortcut mapping through 1×1 convolution. The two branch features are finally concatenated and output, achieving a balance between feature extraction capability and lightweight characteristics. These modules are used as the basic units of Slim Neck, and the structures of the two modules are shown in Figure 9.

In the Neck improvement, all DS-C3k2 modules in the Neck are replaced with VoVGSCSP, all Conv layers in the downsampling stage are replaced with GSConv. The improved Slim Neck further reduces the computational cost of the model while alleviating the channel separation problem caused by depthwise convolution, optimizing multi-scale traffic sign feature fusion and enhancing the feature discrimination capability of occluded and deformed traffic signs.

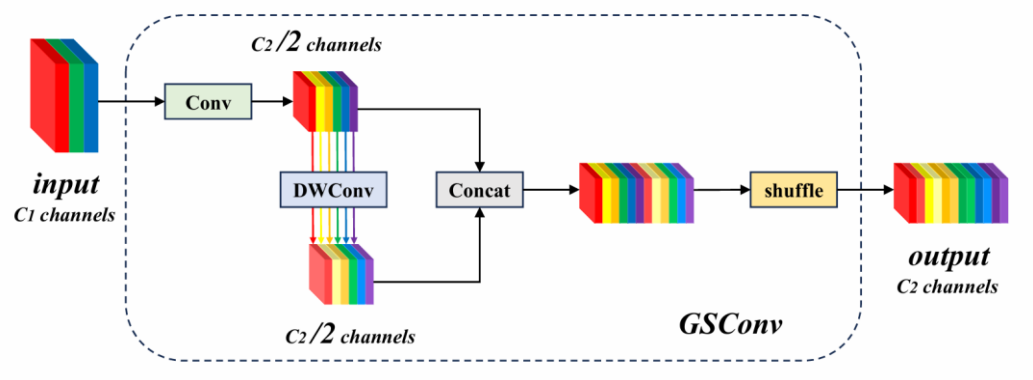

Figure 8: Structure of GSConv.

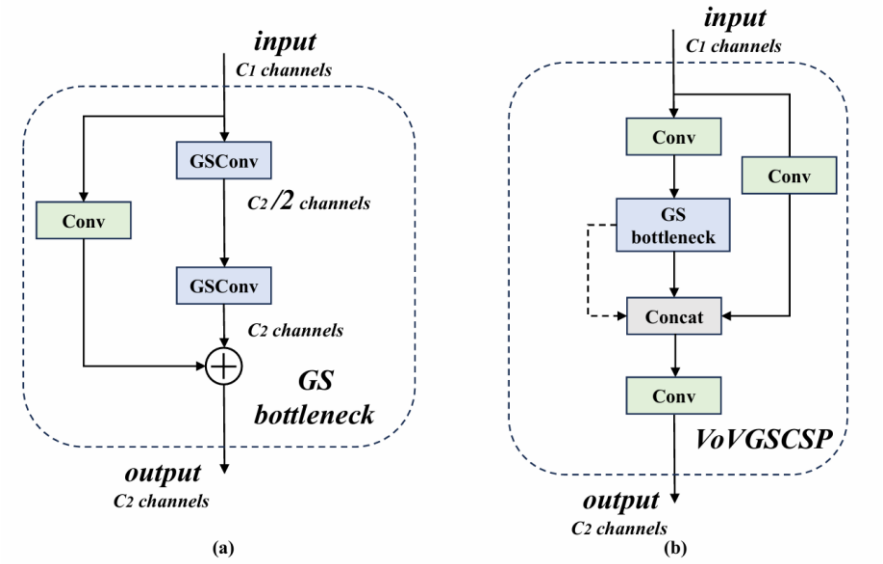

Figure 9: Structure of Lightweight Slim Neck Core Modules. (a) GS Bottleneck Structure; (b) VoVGSCSP Structure.

3.3. Lightweight Detection Head Based on MBConv

The original decoupled Detection Head of YOLOv13n relies on standard convolution to construct the classification and regression branches. However, the overall number of parameters and computational cost still have room for reduction, resulting in certain redundancy when deployed on vehicle-mounted embedded devices with limited computing resources. To further achieve model lightweighting while maintaining stable feature extraction capability, this paper introduces the MBConv module to reconstruct the Detection Head structure.

MBConv, namely Mobile Inverted Bottleneck Convolution, adopts the inverted bottleneck design principle of "expansion first, feature extraction second, and compression afterward". The input feature is first expanded in channel dimension through 1×1 pointwise convolution, enlarging the feature representation space and providing sufficient dimensional support for subsequent feature extraction. Subsequently, depthwise separable convolution is used to perform spatial feature extraction and model spatial information with a very small increase in the number of parameters. An SE channel attention unit is embedded inside the module, which generates channel weights through global average pooling and fully connected layers, adaptively enhancing effective feature channels and suppressing redundant background information. Finally, 1×1 pointwise convolution is used to compress the channel dimension back to the input dimension, achieving feature fusion and parameter compression. When the size and number of channels of the input and output features are completely consistent, the module introduces a shortcut residual connection to reuse shallow features while alleviating the gradient vanishing problem in deep networks and improving training stability.

Based on MBConv, this paper constructs a lightweight decoupled Detection Head and mainly reconstructs the regression branch. The improved structure is shown in Figure 10. The bounding box regression branch stacks two MBConv modules to replace the original Conv layers, followed by an output convolution layer and the DFL distribution integral module to complete the distribution fitting of bounding box coordinates and width-height offsets. The classification branch retains the original depthwise separable convolution structure, maintaining category discrimination capability while controlling computational cost. The improved Detection Head significantly reduces computational and storage overhead while maintaining good feature extraction capability and detection accuracy, further enhancing the deployment adaptability of the model on low-computing-power vehicle-mounted devices.

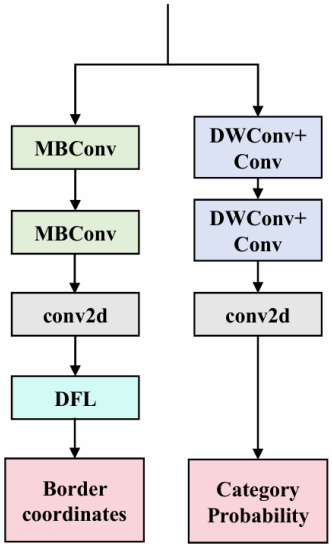

Figure 10: Structure of the Improved Detection Head.

3.4. Bounding Box Regression Loss Function Based on Inner-CIoU

YOLOv13n adopts the CIoU loss function for the original bounding box regression task. This loss function introduces center distance and aspect ratio constraints based on the Intersection over Union (IoU), which can optimize the bounding box regression process to a certain extent. However, CIoU assigns equal loss weights to all training samples and cannot adaptively adjust the optimization strength according to the regression difficulty of different samples. In vehicle-mounted traffic sign detection scenarios, distant small targets and occluded or deformed traffic signs generally belong to low-IoU difficult samples. During training, the loss is easily dominated by a large number of high-IoU clear traffic signs, resulting in insufficient bounding box regression for difficult samples, causing localization deviation and slow convergence, which directly affects the detection accuracy of small targets and occluded targets.

To address this problem, this paper introduces the Inner-CIoU loss function to replace the original CIoU. The core idea of Inner-CIoU is to introduce auxiliary bounding boxes and a scale factor $ratio$, where the size of auxiliary boxes is adjusted to adapt to the regression requirements of different samples. After defining the center coordinates and width-height parameters of the ground-truth box B^{gt} and the predicted box B, the vertex coordinates of two auxiliary bounding boxes are calculated by combining the scale factor $ratio$:

$$\begin{cases} b_l^{gt} = x_c^{gt} - \dfrac{w^{gt} \cdot ratio}{2}, & b_r^{gt} = x_c^{gt} + \dfrac{w^{gt} \cdot ratio}{2} \\ b_t^{gt} = y_c^{gt} - \dfrac{h^{gt} \cdot ratio}{2}, & b_b^{gt} = y_c^{gt} + \dfrac{h^{gt} \cdot ratio}{2} \\ b_l = x_c - \dfrac{w \cdot ratio}{2}, & b_r = x_c + \dfrac{w \cdot ratio}{2} \\ b_t = y_c - \dfrac{h \cdot ratio}{2}, & b_b = y_c + \dfrac{h \cdot ratio}{2} \end{cases} \tag{1}$$

The intersection $inter$, union $union$, and auxiliary Intersection over Union IoU^{inner} are calculated based on the two auxiliary bounding boxes:

$$inter = \left(min\left(b_r^{gt}, b_r\right) - max\left(b_l^{gt}, b_l\right)\right) \cdot \left(min\left(b_b^{gt}, b_b\right) - max\left(b_t^{gt}, b_t\right)\right) \tag{2}$$

$$union = (w^{gt} \cdot h^{gt}) \cdot ratio^2 + (w \cdot h) \cdot ratio^2 - inter \tag{3}$$

$$IoU^{inner} = \frac{inter}{union} \tag{4}$$

The value range of the scaling factor $ratio$ is [0.5,1.5]: for high-IoU clear traffic signs, $ratio = 0.8$ is adopted to reduce the size of auxiliary bounding boxes and accelerate convergence; for low-IoU occluded and small traffic signs, $ratio = 1.2$ is adopted to enlarge the auxiliary bounding boxes and improve the regression performance. By combining the auxiliary Intersection over Union with the original CIoU loss, the expression of the Inner-CIoU loss is obtained:

$$L_{Inner-CIoU} = L_{CIoU} + IoU - IoU^{inner} \tag{5}$$

Compared with the original CIoU, Inner-CIoU achieves sample-adaptive optimization through auxiliary bounding boxes. It not only retains the constraint capability of CIoU on bounding box geometric features but also improves the problems of slow convergence and poor fitting performance on difficult samples in traditional IoU-based losses. After being applied to vehicle-mounted traffic sign detection tasks, the model significantly improves the bounding box localization accuracy for occluded and distant small traffic signs, while the overall training convergence efficiency is also

optimized.

4. Experiments and Results Analysis

4.1. Dataset

The main experiments in this paper are conducted on the TT100K road traffic sign dataset. The images in this dataset are captured from real urban road scenes using vehicle-mounted cameras, covering various real driving environments, including sunny weather, backlighting, rainy conditions, and nighttime scenarios. The samples contain practical characteristics such as distant small targets, partial occlusion, and shape distortion, which closely match real vehicle-mounted detection scenarios. To avoid sample distribution imbalance caused by rare categories, categories with insufficient samples are removed. Finally, 45 traffic sign categories with more than 100 annotated samples per category are retained for model training and evaluation. To ensure a balanced and reasonable sample distribution, all selected samples are randomly shuffled and then divided into the training set, validation set, and test set according to the ratio of 7:2:1, containing 6793, 1949, and 996 images, respectively. In addition, to evaluate the generalization capability of the TS-YOLOv13 model, the CCTSDB, GTSDB, and LISA datasets are additionally introduced for cross-dataset testing.

4.2. Experimental Environment Configuration

The hardware and software configurations required for the experiments are as follows: the operating system is Windows 11, the CPU is Intel Core i7-13700, and the GPU is NVIDIA RTX 4080 with 16 GB of memory. The deep learning framework is PyTorch 2.4.1, the programming language is Python 3.11.15, and the CUDA computing architecture version is 12.1.

The main training hyperparameter settings are presented in Table 1. All experiments use an input image size of 640×640. The optimizer adopts stochastic gradient descent (SGD). During the training process, an early stopping strategy is employed: if the detection accuracy on the validation set does not improve for 30 consecutive epochs, the training is terminated early to prevent model overfitting.

Table 1: Detailed Experimental Parameter Settings.

Parameter	Setting	Parameter	Setting
epoch	200	optimizer	SGD
patience	30	warmup_epochs	3.0
batch	16	momentum	0.937
workers	8	lrf	0.01
imgsz	640	lr0	0.01

The precision (P), recall (R), mean Average Precision at an IoU threshold of 0.5 (mAP@0.5), number of parameters (Params), floating-point computation (GFLOPs), and model size (Size) are selected as the comprehensive evaluation metrics for model performance. Among them, P, R, and mAP@0.5 are used to measure the localization and classification accuracy of traffic sign detection, while Params, GFLOPs, and Size are used to evaluate the lightweight capability and hardware deployment cost of the model.

4.3. Ablation Experiments

To quantitatively analyze the impact of the four improved modules on model performance, YOLOv13n is adopted as the baseline model. Ablation experiments are conducted on the selected 45-category TT100K dataset using a progressive addition strategy. The ShuffleNetV2 lightweight

backbone (SV2), Slim Neck structure (SN), Detect_MBConv lightweight detection head (MBH), and Inner-CIoU loss function (ICIoU) are introduced sequentially. The training environment and hyperparameter settings of all experimental groups are kept exactly the same. In the table, "√" indicates that the corresponding improvement module is enabled. The ablation experiment results are shown in Table 2.

Table 2: Ablation Experiment Results on the TT100K Dataset.

Method	SV2	SN	MBH	ICIoU	P (%)	R (%)	mAP@0.5 (%)	Params (M)	GFLOPs	Size (MB)
V13n	-	-	-	-	60.2	53.2	58.5	2.45	6.2	5.4
A	√	-	-	-	62.0	58.5	60.9	2.38	6.7	5.2
B	√	√	-	-	64.4	57.3	61.5	2.39	6.3	5.3
C	√	√	√	-	64.4	55.1	61.2	2.07	5.1	4.6
D	√	√	√	√	70.5	57.0	63.8	2.07	5.1	4.6

The ablation experiment results show that replacing the backbone with ShuffleNetV2 reduces Params and Size while improving all detection metrics, with only a slight increase in GFLOPs caused by channel shuffle. On this basis, introducing the Slim Neck structure further reduces GFLOPs, and improves P and mAP@0.5, optimizing the multi-scale feature fusion capability for occluded and distorted traffic signs. The introduction of the Detect_MBConv lightweight detection head significantly reduces Params, GFLOPs, and Size, with only slight decreases in R and mAP@0.5, achieving lightweight benefits at the cost of minor accuracy degradation. Finally, adding the Inner-CIoU loss function introduces no additional computational or parameter overhead, while significantly improving P and enhancing the bounding box regression performance for difficult samples. The proposed TS-YOLOv13 with four improvement strategies achieves better performance than the original YOLOv13n. Specifically, P increases from 60.2% to 70.5%, R increases from 53.2% to 57.0%, and mAP@0.5 increases from 58.5% to 63.8%. Meanwhile, Params decreases from 2.45M to 2.07M, GFLOPs decreases from 6.2 to 5.1, and Size decreases from 5.4MB to 4.6MB. Therefore, TS-YOLOv13 achieves comprehensive improvements in detection accuracy while reducing model complexity, balancing inference performance and deployment cost, and is more suitable for vehicle-mounted embedded application scenarios.

4.4. Comparison Experiments

Table 3: Comparison Results on the TT100K Dataset.

Model	P (%)	R (%)	mAP@0.5 (%)	Params (M)	GFLOPs	Size (MB)
YOLOv3-tiny	71.5	47.8	54.0	12.15	19.0	24.4
YOLOv5n	62.3	57.3	62.2	2.51	7.1	5.3
YOLOv6n	52.9	54.7	55.7	4.29	12.0	8.8
YOLOv8n	68.3	55.4	61.7	3.01	8.1	6.3
YOLOv10n	60.3	57.2	60.1	2.71	8.3	5.8
YOLOv11n	60.6	56.1	59.7	2.59	6.4	5.5
YOLOv12n	63.4	53.5	58.1	2.52	5.9	5.5
YOLOv13n	60.2	53.2	58.5	2.45	6.2	5.4
TS-YOLOv13	70.5	57.0	63.8	2.07	5.1	4.6

To objectively evaluate the overall performance of TS-YOLOv13, several mainstream lightweight object detection algorithms are selected for comparison. All models are trained and tested on the

selected 45-category TT100K dataset under the same experimental settings. The comparison models include YOLOv3-tiny, YOLOv5n, YOLOv6n, YOLOv8n, YOLOv10n, YOLOv11n, YOLOv12n, and YOLOv13n. The experimental results are summarized in Table 3.

The comparison results show that TS-YOLOv13 achieves the lowest Params, GFLOPs, and Size among all lightweight models, demonstrating significant advantages in lightweight design. Meanwhile, its mAP@0.5 reaches 63.8%, outperforming all comparison algorithms. Compared with mainstream lightweight YOLO models of different generations, the proposed method achieves a better overall balance among detection accuracy, inference speed, and deployment cost, making it more suitable for vehicle-mounted embedded detection scenarios with limited resources.

4.5. Generalization Experiments

To verify the effectiveness and generalization capability of TS-YOLOv13 for small object detection, generalization tests are conducted on the CCTSDB, GTSDB, and LISA datasets, respectively. The experimental results are presented in Table 4.

Table 4: Comparison Results on Different Datasets.

Dataset	Model	P (%)	R (%)	mAP@0.5 (%)	Params (M)	GFLOPs	Size (MB)
CCTSDB	YOLOv13n	91.8	86.1	92.6	2.45	6.2	5.4
CCTSDB	TS-YOLOv13	94.2	91.3	95.6	2.06	5.1	4.6
GTSDB	YOLOv13n	79.8	74.9	78.7	2.45	6.2	5.4
GTSDB	TS-YOLOv13	84.2	78.8	80.8	2.06	5.1	4.6
LISA	YOLOv13n	85.6	80.9	82.9	2.45	6.2	5.4
LISA	TS-YOLOv13	88.9	83.2	86.8	2.06	5.1	4.6

The multi-dataset test results demonstrate that TS-YOLOv13 maintains stable and reliable detection performance across traffic sign datasets with different scenarios and acquisition standards, exhibiting strong cross-scenario feature adaptation capability and generalization performance. It achieves reliable detection performance under different illumination conditions, object scales, and traffic scenarios, further demonstrating the effectiveness and general applicability of the proposed improvement strategies.

4.6. Visualization of Results

To visually demonstrate the effectiveness of the improved model, Figure 11 compares the four key metrics, namely P, R, mAP@0.5, and Val Box Loss, between YOLOv13n and TS-YOLOv13 on the TT100K dataset. The comparison results show that TS-YOLOv13 outperforms the original YOLOv13n in all four metrics. Specifically, P and R increase more rapidly and converge to higher final values, mAP@0.5 achieves the most significant improvement, and Val Box Loss decreases faster and converges to a lower value, indicating superior localization accuracy and generalization performance.

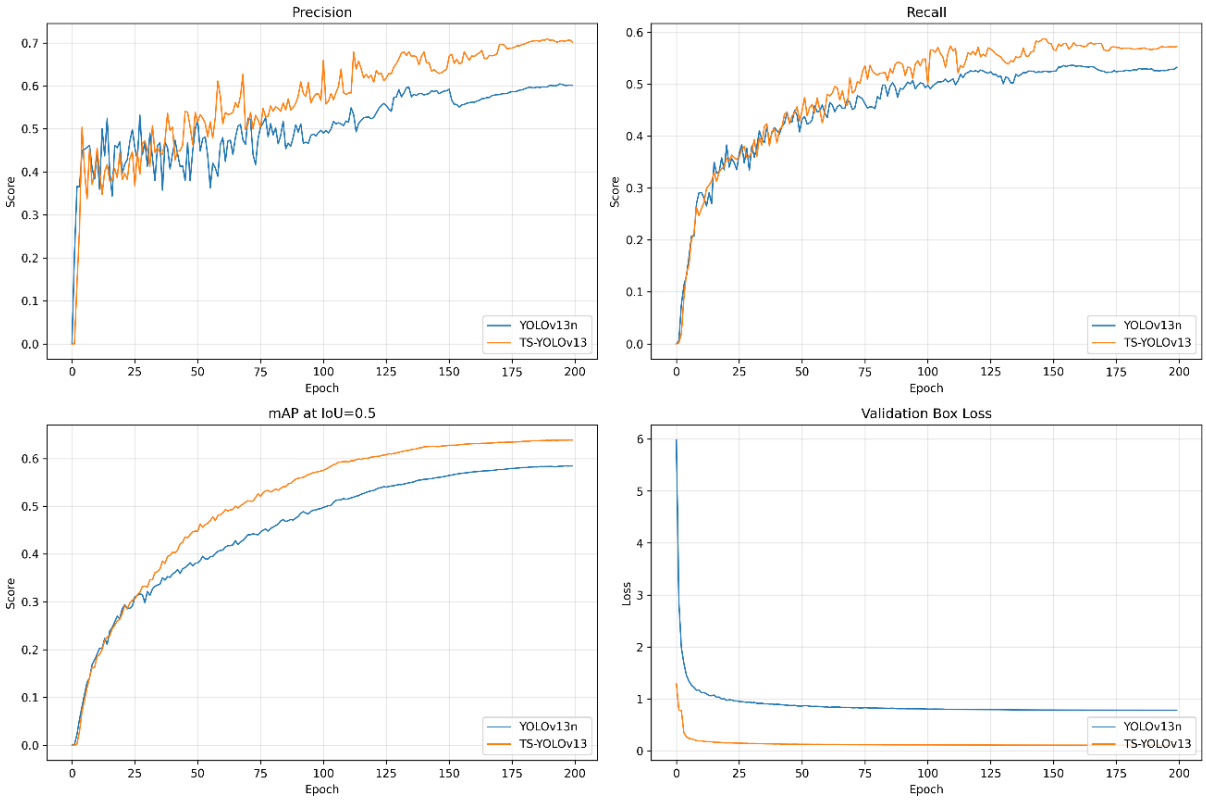

Figure 11: Comparison of Different Performance Metrics.

5. Conclusion

This paper proposes a lightweight traffic sign detection algorithm, TS-YOLOv13, based on YOLOv13n to address the problems of missed detection of small targets, insufficient adaptability to complex environments, and limited model deployment in vehicle-mounted embedded scenarios. The proposed algorithm performs collaborative optimization of the backbone network, feature fusion, detection head structure, and loss function. While effectively reducing the number of model parameters and computational complexity, it improves the feature representation capability and localization accuracy of traffic signs in small-target and occluded scenarios. Ablation experiments and comparison experiments are conducted on the TT100K dataset. In addition, the cross-scenario generalization capability of the model is further validated on the CCTSDB, GTSDB, and LISA datasets. The experimental results demonstrate that TS-YOLOv13 achieves good overall performance in terms of detection accuracy, lightweight design, and inference efficiency, meeting the application requirements of real-time traffic sign detection on vehicle-mounted embedded platforms. Future work will further focus on feature enhancement and model compression in complex road environments. On the basis of maintaining detection accuracy, the network inference efficiency will be optimized to improve the practical deployment capability of the proposed algorithm on low-computing-power edge devices.

References

[1] Liu L P. Improved canny edge detection algorithm matches traffic signs[J]. Proceedings of SPIE, 2016, 10033.
[2] Setiyono B, Wijaya R N R, Sulistyaningrum D R, et al. Vehicle counting and classification in varying traffic conditions using Faster R-CNN[J]. Journal of Physics: Conference Series, 2025, 2942(1): 012025.
[3] Omodaratan B, Jamali A, Wiley T, et al. Advances in You Only Look Once (YOLO) algorithms for lane and object

detection in autonomous vehicles[J]. Engineering Applications of Artificial Intelligence, 2026, 168: 113893.
[4] Han J, Wang C, Du J. HMS-YOLO: Multi-scale traffic sign detection for complex scenarios[J]. Digital Signal Processing, 2026, 183: 106370.
[5] Ma S, Zhao N, Li J, et al. MFSF-YOLO: An improved YOLO11 model for traffic sign detection and recognition[J]. Engineering Letters, 2026, 34(6).
[6] Dong W Y, Chen Y X, Zou G Y. Autotrinet YOLO triple attention framework for robust traffic sign detection[J]. Scientific Reports, 2025, 15(1): 42247.
[7] Xu J, Du Y, Yi Y, et al. An improved lightweight algorithm for traffic sign detection[J]. Scientific Reports, 2025, 15(1): 33554.
[8] Guo S, Zhao N, Ouyang X, et al. RBL-YOLOv8: A lightweight multi-scale detection and recognition method for traffic signs[J]. Engineering Letters, 2024, 32(11).
[9] Zhang J, Yi Y, Wang Z, et al. Learning multi-layer interactive residual feature fusion network for real-time traffic sign detection with stage routing attention[J]. Journal of Real-Time Image Processing, 2024, 21(5): 176.
[10] Zhang X, Tian Y. Improved YOLOv5s traffic sign detection[J]. Engineering Letters, 2023, 31(4).
[11] Han T, Sun L, Dong Q. An improved YOLO model for traffic signs small target image detection[J]. Applied Sciences, 2023, 13(15).
[12] Li D, Hu Y, Cao W, et al. A lightweight fabric defect detection method based on improved ShuffleNetV2[J]. Journal of Real-Time Image Processing, 2026, 23(1): 44.
[13] Huang L, Lai X, Lin P, et al. Lightweight vehicle damage detection using GSConv-based Slim-Neck and bi-level routing attention[J]. World Electric Vehicle Journal, 2026, 17(6): 290.