technologies

MDPI

Article

Research on Traffic Sign Detection Algorithm Based on Improved YOLO11n

Haonan Feng [1], Jiaxu Meng [2], Zhiyong Guo [1,*], Pengchao Zhao [2], Wenchao Zhang [2], Yiran Cao [1] and Cunman Liang [3]

[1] Aeronautical Engineering Institute, Civil Aviation University of China, Tianjin 300300, China; haon_f@163.com (H.F.)
[2] China Automotive Technology & Research Center Co., Ltd., Tianjin 300300, China
[3] School of Mechanical Engineering, Tianjin University, Tianjin 300350, China
* Correspondence: zyguo@cauc.edu.cn

Abstract

In order to improve detection accuracy while minimizing computational overhead, a modified algorithm is proposed based on the YOLO11n baseline. The innovation incorporates a lightweight ADown module into the P4 and P5 layers of the backbone network, strategically reducing computational complexity. Simultaneously, a multi-scale attention mechanism with parallel structure is integrated into the detection head to enhance feature representation, while a micro-detection head is appended to specifically improve the detection of tiny objects. Based on the classic metrics, including parameter count, mAP@50, mAP@50-95, recall, and FPS, the ablation experiments are performed to validate the improvement of the improved algorithm on the CCTSDB2021 dataset. Furthermore, comparative experiments against traditional YOLO variants are conducted on both CCTSDB2021 and TT100K-2021 datasets. Experimental results demonstrate significant improvements across all evaluated metrics for the improved algorithm, highlighting its exceptional capability to balance high accuracy with minimal computational complexity.

Keywords: YOLO11n; traffic sign detection; attention mechanisms; lightweight model; detection head

1. Introduction

With the rapid development of autonomous driving technology, accurate and efficient traffic sign detection has become increasingly crucial [1–3]. Traffic signs provide essential road information and support intelligent driving systems in route planning, speed regulation, and safety navigation. However, complex road environments, large variations in sign scale and appearance, and adverse conditions such as illumination changes and bad weather still pose major challenges to robust, real-time detection.

Deep-learning–based object detection algorithms, particularly convolutional neural network (CNN)-based methods, have greatly advanced traffic sign detection. Two-stage detectors such as R-CNN, Fast R-CNN, and Faster R-CNN [4–6] achieve high accuracy but are often too computationally expensive for real-time deployment. One-stage detectors, represented by YOLO [7–9] and SSD [10], directly predict object categories and locations from feature maps in a single pass, offering a better speed–accuracy trade-off, making them attractive for intelligent vehicles. Among them, the YOLO series has been widely adopted in traffic sign detection due to its balance between real-time performance and detection precision.

Check for updates

Academic Editor: Pedro Antonio Gutiérrez

Received: 15 September 2025
Revised: 17 December 2025
Accepted: 19 December 2025
Published: 21 December 2025

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

To further improve the YOLO network, many studies have modified its backbone, neck, and head to enhance small-object detection or reduce computational cost. For instance, YOLOv7 [11] and YOLOv8 [12] redesign feature-extraction and detection heads to improve multi-scale representation and inference efficiency. Other works introduce lightweight modules or tailored feature fusion strategies to better handle small traffic signs or resource-constrained platforms, such as YOLO-ADual [13], DP-YOLO [14], and methods based on depthwise separable convolutions, optimized loss functions, or knowledge distillation [15–18]. Nevertheless, there is still a lack of approaches that simultaneously preserve the lightweight characteristics of the YOLO series and systematically enhance feature representation and multi-scale detection performance through coordinated modular optimization.

Motivated by this gap, this study proposes an improved traffic sign detection algorithm based on YOLO11n that aims to enhance accuracy while maintaining a light model: (1) The proposed network integrates an ADown module into the backbone to reduce computational cost while preserving feature quality; (2) It introduces a high-resolution feature layer coupled with a micro-detection head to strengthen the perception of small traffic signs; and (3) It designs a Multi-Scale Convolutional Block Attention Module (MSCBAM) to refine multi-scale feature representation. Rather than adding isolated modules, the network performs coordinated optimization across the feature extraction layers (P4/P5), feature fusion layers, and detection heads to achieve a globally improved lightweight architecture.

The remainder of this paper is organized as follows. Section 2 details the architecture of the proposed network and its component enhancements. Section 3 describes the experimental setup on the CCTSDB2021, TT100K-2021, and Visdrone2019 datasets. Section 4 presents comprehensive experimental results and comparisons with mainstream algorithms in terms of precision, recall, and efficiency. Section 5 concludes the paper.

2. YOLO11n Network Structure Optimization

Based on the traditional YOLO11n network architecture, this study proposes an improved network structure. As shown in Figure 1, the conventional convolutional modules in the P4 and P5 layers of the YOLO11n backbone are replaced with Adaptive Downsampling (ADown) modules to enhance the network's lightweight design. A micro-detection head is incorporated at the output end, introducing a high-resolution feature layer to improve the detection accuracy of tiny objects and a Multi-Scale Convolutional Block Attention Module is proposed and applied to the medium, small, and micro-detection heads to enhance detection accuracy.

2.1. Design of the ADown Lightweight Module

The ADown module is a lightweight feature downsampling module, which has been designed to efficiently extract and fuse key features by using a multi-branch strategy while reducing computational complexity [19]. As shown in Figure 2, the ADown module first takes an input feature map $X \in \mathbb{R}^{H\times W\times C}$ and applies a 2×2 average pooling operation with stride 1 to obtain a smoothed feature map:

$$X' = \text{AvgPool2D}(X; k = 2, s = 1, p = 0), \quad X' \in \mathbb{R}^{(H-1)\times(W-1)\times C}. \tag{1}$$

Then, X' is evenly split along the channel dimension into two sub-feature maps X_1 and X_2, each containing half of the original channels:

$$X_1, X_2 = \text{Split}(X', \text{channel}), \quad X_1, X_2 \in \mathbb{R}^{(H-1)\times(W-1)\times \frac{C}{2}}. \tag{2}$$

Channel splitting helps to reduce the computational load and memory usage of each individual branch. For the X_1 branch, a 3×3 convolution is employed to extract local spatial features and perform downsampling:

$$\tilde{X_1} = \text{Conv}(X_1, k = 3, s = 2, p = 1), \quad \tilde{X_1} \in \mathbb{R}^{\frac{H}{2} \times \frac{W}{2} \times \frac{C}{2}}. \tag{3}$$

For the X_2 branch, a max pooling operation is first used to retain critical responses, followed by a 1×1 convolution to achieve channel compression and feature fusion:

$$\tilde{X_2} = \text{MaxPool2D}(X_2, k = 3, s = 2, p = 1), \quad \tilde{X_2} \in \mathbb{R}^{\frac{H}{2} \times \frac{W}{2} \times \frac{C}{2}}. \tag{4}$$

$$\tilde{\tilde{X_2}} = \text{Conv}(\tilde{X_2}, k = 1, s = 1, p = 0), \quad \tilde{\tilde{X_2}} \in \mathbb{R}^{\frac{H}{2} \times \frac{W}{2} \times \frac{C}{2}}. \tag{5}$$

Finally, the outputs of the two branches are concatenated along the channel dimension to form the output feature map of the ADown module. Replacing parameter-intensive convolutional layers with parameter-free pooling operations can significantly enhance inference efficiency and minimize model footprint. The removal of residual connections and attention mechanisms eliminates additional computational branches, rendering the model more suitable for lightweight architectures and edge device deployment. The output feature map is subsequently formed by concatenating the two branches along the channel dimension. In the course of feature extraction, the ADown module captures multi-scale information through an efficient convolutional structure, enabling the downsampled features to maintain sufficient discriminative capability. In terms of lightweight design, the ADown module significantly reduces the computational load and memory usage of each path by splitting the feature map along the channel dimension. It also replaces some parameterized convolution layers with parameter-free operations such as average and max pooling, thereby improving inference speed and reducing model size. Furthermore, the use of small convolution kernels (1×1 and 3×3) instead of larger ones effectively diminishes the number of floating-point operations.

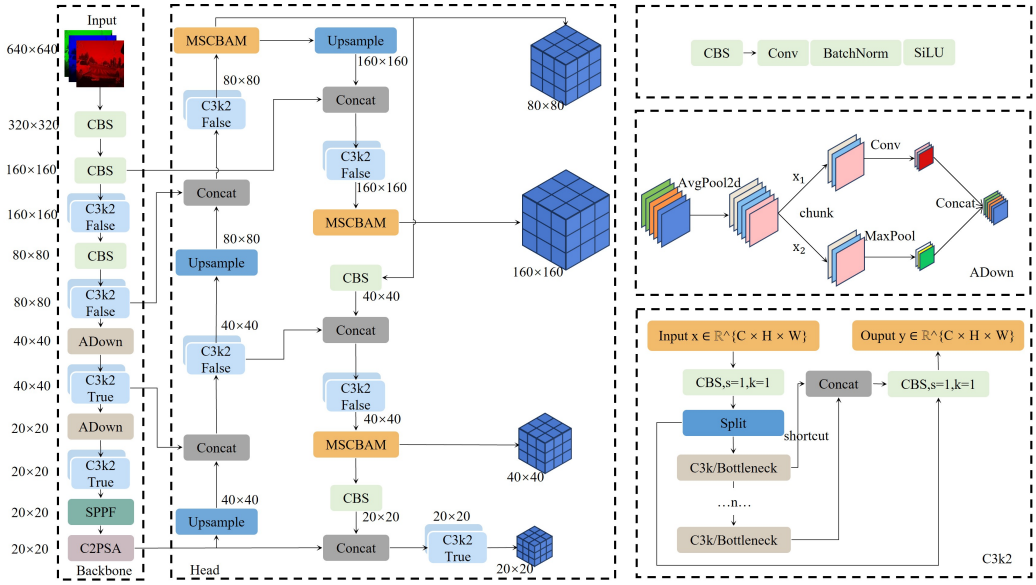

Figure 1. Structure of the improved network.

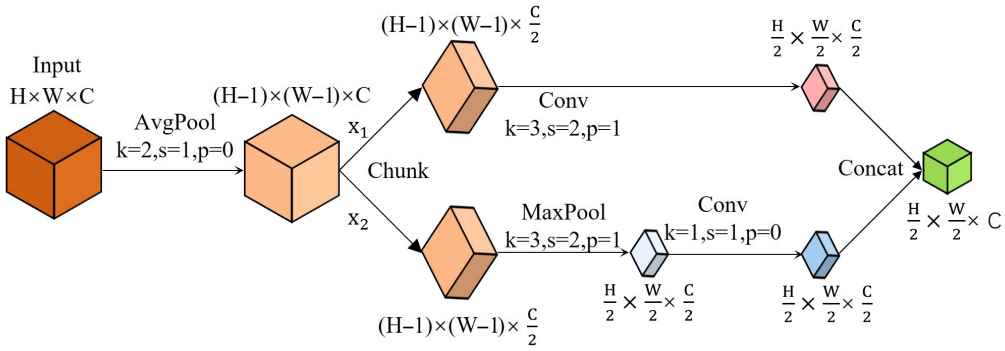

Figure 2. Structure of ADown.

2.2. Design of the Multi-Scale Convolutional Block Attention Module

The proposed Multi-Scale Convolutional Attention Mechanism is depicted in Figure 3a. It employs two connection strategies: a parallel structure and a sequential structure. In the parallel structure, input features are independently routed to the Channel Attention (CA) and Spatial Attention (SA) modules. The outputs of these modules are weighted by coefficients k_1 and k_2 and fused via weighted summation to produce the final output.

In the sequential structure, the input features are first processed by the CA module to generate intermediate features, which are subsequently passed through the SA module. The final output is obtained by weighting the outputs of CA and SA with k_1 and k_2 and combined to form the final output.

As shown in Figure 3b, the CA module obtains global information along the channel dimension through global average pooling; this operation's formulation is defined as follows:

$$F_{\text{avg}}^{(c)} = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} X_{c,i,j} \tag{6}$$

Subsequently, two successive 1×1 convolutional layers are applied to first reduce and then restore the feature dimensionality. A Sigmoid activation function is employed to generate the Channel Attention map. The resulting channel weights are normalized to the range [0, 1] and used to perform element-wise multiplication with the original input feature map on a per-channel basis.

As illustrated in Figure 3c, the SA module first enhances the model's ability to perceive salient regions and global distribution by applying average pooling and max pooling across the channel dimension. The resulting two spatial feature maps are concatenated and passed through a multi-scale convolutional module with kernels of sizes 3×3, 5×5, and 7×7 to capture spatial features at different scales. A 1×1 convolution is then used to fuse these multi-scale features, followed by the application of a SiLU activation function, which highlights critical regions while suppressing less informative ones, thereby improving the model's representational power. The corresponding formulations are given below:

$$\text{SiLU}(x) = x \cdot \sigma(x) = \frac{x}{1 + e^{-x}} \tag{7}$$

$$X_{\text{avg}}(n, 1, h, w) = \frac{1}{C} \sum_{i=1}^{C} X(n, i, h, w) \tag{8}$$

$$X_{\text{max}}(n, 1, h, w) = \max_{1 \le i \le C} X(n, i, h, w) \tag{9}$$

The outputs of CA and SA are computed independently and fused through weighted summation, thereby balancing the importance of channel-wise and spatial information to generate the final enhanced feature map. We assign weights k_1 and k_2 to the Channel and

Spatial Attention modules and determine the optimal weight ratio through experiments, which are discussed in detail in Section 4.1.

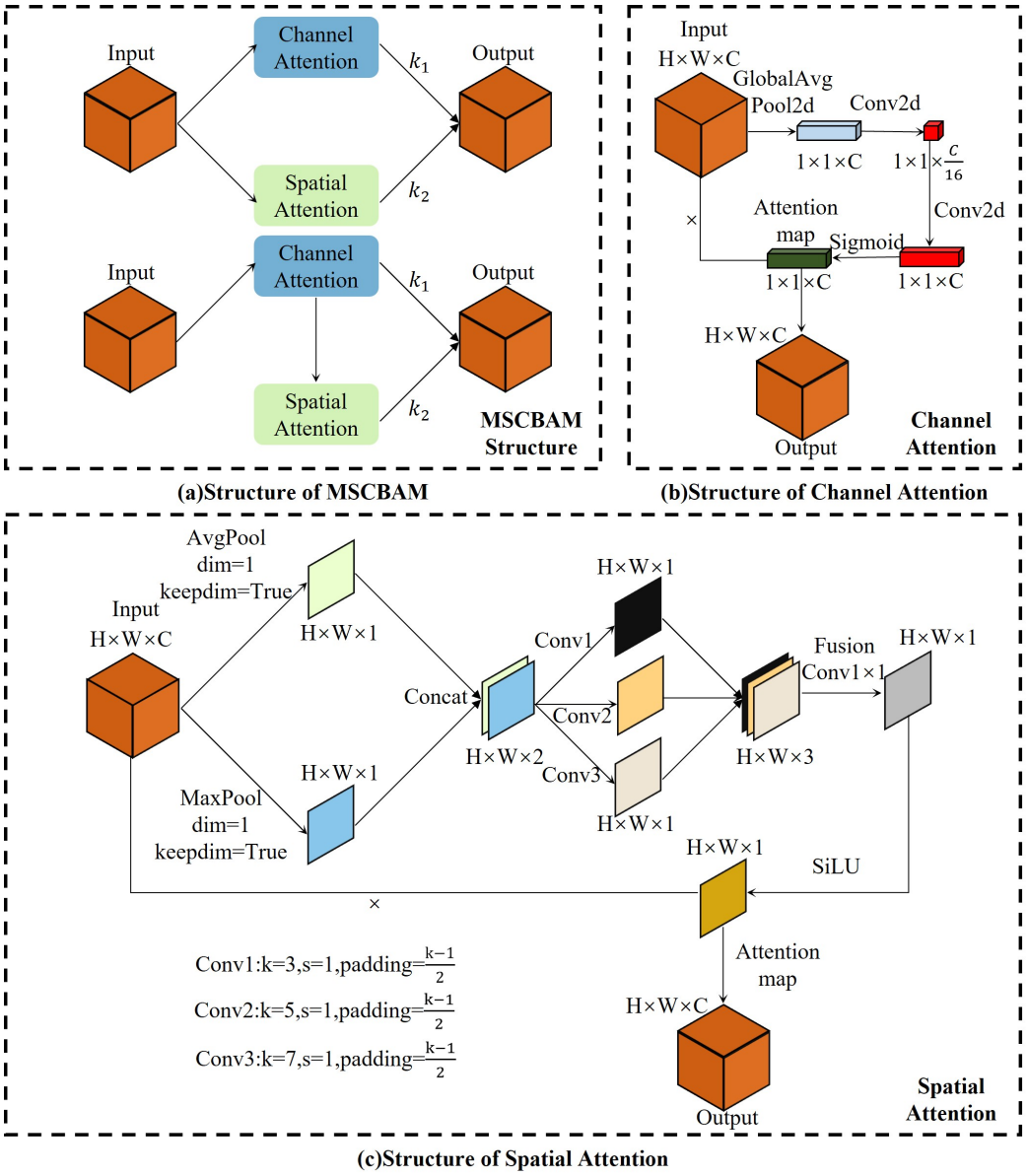

Figure 3. Structure of MSCBAM.

2.3. High-Resolution Feature Layer with a Micro-Detection Head

Tiny objects often occupy only a few pixels in the original image, making their features highly susceptible to loss during successive downsampling operations. To improve detection accuracy, it is essential to construct high-resolution feature layers that preserve more spatial detail and facilitate the extraction of fine-grained features such as edges and shapes. This is particularly advantageous in tiny object detection tasks, as demonstrated on the CCTSDB2021 dataset, where high-resolution features contribute significantly to improved model performance.

In conventional convolutional neural networks, downsampling during feature extraction expands the receptive field and captures high-level semantics information. However, this process often sacrifices local spatial details, which is critical for accurately recognizing tiny objects. The proposed algorithm addresses this issue by introducing a micro-detection head that explicitly targets tiny objects.

Specifically, in the original YOLO11n architecture, the feature maps fed into the detection heads have spatial resolutions of 20 × 20, 40 × 40, and 80 × 80, corresponding to large, medium, and small objects, respectively. To further enhance the detection performance for tiny objects, the improved algorithm introduces an additional 160 × 160 branch in the head network. This micro-detection head follows three core design principles:

- *High-resolution representation*: By generating a 160 × 160 feature map, the model retains finer spatial details, enabling the network to capture edges, contours, and other microstructures that are often lost in low-resolution heads.
- *Multi-level feature fusion*: The branch incorporates both shallow-layer features rich in spatial detail and upsampled deeper features that carry semantic context. This fusion mechanism ensures that the head simultaneously leverages fine-grained geometry and high-level semantics.
- *Attention-guided refinement*: The fused features are processed by the proposed Multi-Scale Convolutional Attention Module, which adaptively highlights discriminative regions across scales while suppressing background noise, thus improving localization accuracy for tiny objects.

2.4. Overall Pipeline of the Proposed Framework

To provide a clear overview of the entire experimental procedure, the overall pipeline of the proposed traffic sign detection framework is illustrated in Figure 4. First, all images from the TT100K_2021 and CCTSDB2021 datasets are resized to 640 × 640 and augmented using Mosaic. The processed images are then fed into the improved YOLO11n network, which integrates the ADown module, the MSCBAM attention mechanism, and a micro-detection head. After prediction, non-maximum suppression (NMS) is applied to obtain the final traffic sign detection results. Finally, the performance is evaluated in terms of mAP@50, mAP@50–95, recall, inference speed (FPS), and model complexity.

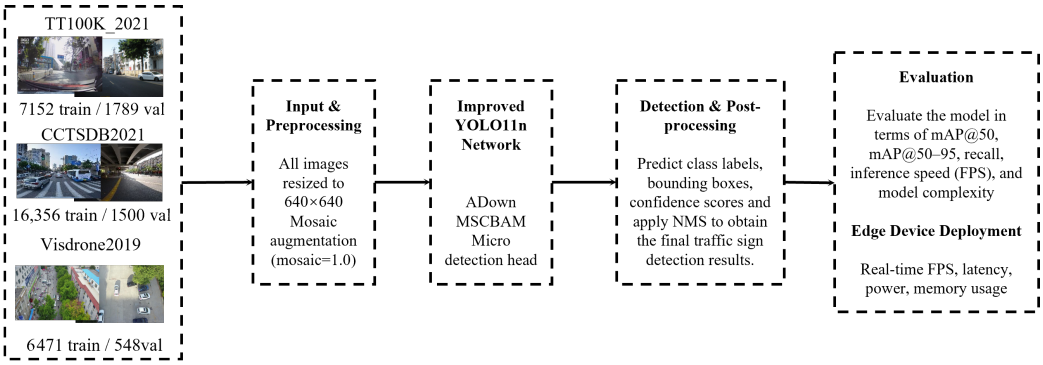

Figure 4. Overall pipeline of the proposed traffic sign detection framework.

3. Experimental Setup

3.1. Experimental Environment

All experiments were performed on a Windows system with 16GB RAM, an NVIDIA RTX 4060 GPU (8 GB), and an Intel Core i7-13700F CPU, using the PyTorch framework. The YOLO-based network was trained for 300 epochs on the CCTSDB2021 dataset and 400 epochs on the TT100K-2021 dataset, with a 3-epoch warm-up. The initial learning rate was 0.01, weight decay 0.0005, momentum 0.937, batch size 16, and input resolution 640 × 640.

3.2. Dataset

The training and validation of the improved network were performed using the CCTSDB2021 dataset [20–22] and the TT100K-2021 dataset [23]. The CCTSDB2021 training set includes 16,356 static images, with 70.49% small and 29.49% medium objects. Object sizes range from 6.0 × 9.0 to 491.4 × 190.0 pixels. The validation set contains 1500 images, with 77.73% small and 22.26% medium objects, ranging from 5.0 × 6.0 to 132.0 × 122.0 pixels. Categories include mandatory, prohibitory, and warning signs.

The TT100K-2021 dataset consists of 8941 images from 45 categories (each with over 100 samples), selected from a larger set of 2048 × 2048 images. After shuffling, data was split 8:2 into 7152 training and 1789 validation images, ensuring category balance. Mosaic augmentation (Mosaic = 1.0) was used during preprocessing to improve generalization. All experiments based on the TT100K-2021 dataset were conducted on this uniformly split dataset to ensure consistency across evaluations.

3.3. Evaluating Indicator

To comprehensively evaluate the performance of the improved network in object detection tasks, the adopted evaluation metrics include mAP@50, mAP@50–95, recall, parameter count, and FPS.

mAP@50 refers to the mean Average Precision over all object categories when the Intersection over Union (IoU) threshold is set to 0.5. AP_i denotes the Average Precision for the i-th category at an IoU threshold of 0.5. It is computed as the area under the corresponding precision–recall (P–R) curve, which is obtained by ranking detections of class i by confidence score and varying the confidence threshold. Accordingly, mAP_{50} is defined as the arithmetic mean of AP_i over all N categories. mAP@50–95 is the average of mean Average Precision values calculated at IoU thresholds ranging from 0.5 to 0.95 with a step of 0.05, i.e., at IoU = (0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95). This metric evaluates the consistency and robustness of the model across varying levels of localization precision. Recall measures the proportion of actual positive samples (true objects) that are correctly detected by the model. A higher recall indicates fewer missed detections. FPS indicates the number of image frames that a model can process per second. A higher FPS value signifies faster image processing speed and more responsive performance. Parameter count reflects the model's complexity and impacts memory usage and inference efficiency. The specific definitions of the evaluation metrics are given by the following formulas:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{10}$$

$$\text{mAP}_{50} = \frac{1}{N} \sum_{i=1}^{N} AP_i \tag{11}$$

$$\text{mAP}_{50-95} = \frac{1}{10} \sum_{i=0}^{9} AP_{\text{IoU}=0.5+0.05i} \tag{12}$$

$$\text{FPS} = \frac{\text{Total Number of Frames}}{\text{Total Inference Time (in seconds)}} \tag{13}$$

TP (True Positives): Number of correctly predicted object instances. FP (False Positives): Number of incorrectly predicted object instances (non-object predicted as object). FN (False Negatives): Number of missed object instances (objects not detected). These metrics provide a comprehensive assessment of detection accuracy, completeness, and model efficiency.

4. Experimental Results and Discussion

4.1. Connection Strategies and Weight Analysis of Channel and Spatial Attention

To evaluate the impact of CA and SA, this study compares detection results using various weight ratios on the CCTSDB2021 dataset, as shown in Tables 1 and 2. When the weights are set to $k_1 = 0.7$ for CA and $k_2 = 0.3$ for SA and the connection strategy is Sequential, the improved network achieves the highest values in recall, mAP@50, and mAP@50–95. This configuration yields the best detection performance, and, thus, in all subsequent experiments, the weights for Channel and Spatial Attention are fixed at 0.7 and 0.3, respectively.

Table 1. Detection results of different Channel Attention and Spatial Attention weights with sequential integration.

Weight Ratio (Channel:Spatial)	Recall (%)	mAP@50 (%)	mAP@50-95 (%)
0.9:0.1	74.22	81.74	53.72
0.8:0.2	71.17	81.08	52.95
0.7:0.3	**75.22**	**82.92**	**54.13**
0.6:0.4	72.07	80.84	53.13
0.5:0.5	74.95	82.12	53.52

Table 2. Detection results of different Channel Attention and Spatial Attention weights with parallel integration.

Weight Ratio (Channel:Spatial)	Recall (%)	mAP@50 (%)	mAP@50-95 (%)
0.9:0.1	74.69	81.72	53.13
0.8:0.2	74.59	**81.86**	53.47
0.7:0.3	74.50	81.43	52.92
0.6:0.4	74.73	81.85	**53.55**
0.5:0.5	**74.98**	81.27	53.14

4.2. Ablation Experiments

For assessing the effectiveness of the improved network, an ablation study was conducted using the YOLO11n baseline under identical conditions. As shown in Table 3, Group 1 represents the original YOLO11n model as the control group.

In Groups 2, 3, and 4, the ADown module, MSCBAM module, and micro-detection head were introduced separately. The results show that mAP@50 increased by 0.2%, 0.22%, and 2.23%, respectively; mAP@50–95 improved by 0.46%, 0.6%, and 2.6%; and recall increased by 0.2%, 1.02%, and 3.14%, respectively. These results demonstrate that all three modules contribute to improved detection accuracy, with the micro-detection head having the most significant impact. From the comparison between Groups 1 and 2, it is evident that the ADown module effectively reduces the number of parameter count, enabling a more lightweight object detection framework. Although the micro-detection head enhances the model's ability to capture fine-grained object details, it also increases parameter count, as seen from the comparison between Groups 1 and 4.

In Groups 5–8, various combinations of the modules were applied. The results indicate that the configuration incorporating all three components—ADown, MSCBAM, and the micro-detection head—achieved the highest detection performance, with mAP@50, mAP@50–95, and recall reaching 82.92%, 54.13%, and 75.22%. Compared to the baseline YOLO11n, these metrics improved by 2.56%, 2.35%, and 2.59%, respectively.

The parameter count in Group 8 is approximately 2.2 M, significantly lower than that in Group 7, indicating that, while the micro-detection head introduces computational over-

head, the ADown module effectively offsets its impact on inference speed. Moreover, the improved algorithm exhibits a noticeably higher FPS compared to the original YOLO11n.

In conclusion, the proposed network achieves an optimal balance between detection accuracy and speed, validating its practical applicability.

Table 3. Ablation experiment on the improved network.

Group	ADown	MSCBAM	Micro-Head	Parameter Count	mAP@50 (%)	mAP@50-95 (%)	Recall (%)	FPS
1				2,590,425	80.36	51.78	72.63	141.62
2	✓			**2,196,953**	80.56	52.24	72.83	151.55
3		✓		2,593,527	80.58	51.67	73.65	**154.57**
4			✓	2,665,900	82.59	53.92	75.77	142.29
5	✓	✓		2,200,055	81.01	52.48	73.66	144.70
6	✓		✓	2,272,428	81.37	52.94	74.29	147.53
7		✓	✓	2,669,333	81.49	53.28	**75.88**	130.34
8	✓	✓	✓	2,267,861	**82.92**	**54.13**	75.22	150.51

4.3. Comparative Experiments

In order to further demonstrate the advantages of the improved network in detection accuracy and model efficiency, it was compared with YOLO11n and three mainstream models—YOLOv5n, YOLOv8n, and YOLOv12n. As shown in Table 4, the proposed network achieved the highest detection accuracy among all methods.

Specifically, compared to YOLOv5n, mAP@50, mAP@50–95, and recall improved by 7.32%, 6.22%, and 7.06%, respectively. Although the proposed network has slightly more parameters than YOLOv5n, it contains 0.7 M and 0.3 M fewer parameters than YOLOv8n and YOLO11n, clearly validating the effectiveness of the ADown module in reducing model complexity. As shown in Figures 5–7, the improved network also outperforms the other models in both detection accuracy and speed.

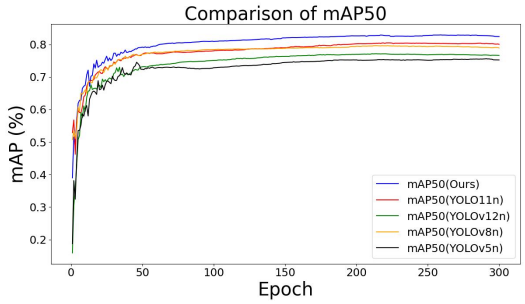

Figure 5. Comparison of mAP50 between the improved network and the original YOLO models.

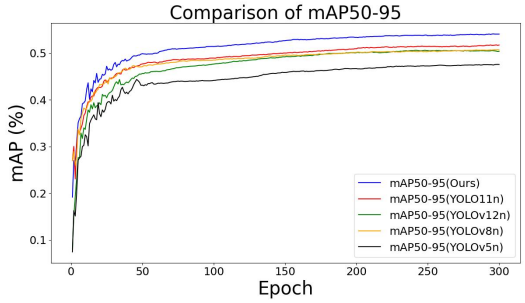

Figure 6. Comparison of mAP50-95 between the improved network and the original YOLO models.

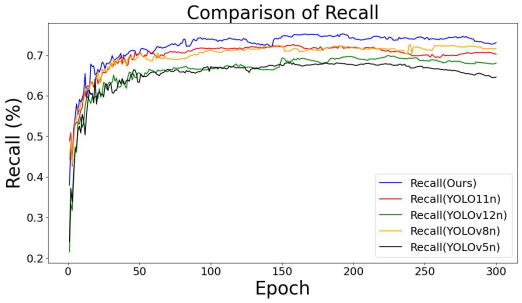

Figure 7. Comparison of recall between the improved network and the original YOLO models.

Based on the CCTSDB2021 dataset, a detailed analysis of the detection performance of the four models was conducted across three traffic sign categories: mandatory, prohibitory, and warning. The mAP@50 and mAP@50–95 results are presented in Table 5. For all three categories, the improved network consistently achieved the highest detection accuracy, with mAP@50 values of 75.7%, 84.1%, and 87.6% and mAP@50–95 values of 52.2%, 56.2%, and 54.0%, respectively. These results indicate that the improved network significantly improves the extraction of tiny object feature information compared to the other algorithms.

Table 4. Comparison of training results before and after improvement of YOLO11n.

Model	Parameter Count	Recall (%)	mAP@50 (%)	mAP@50-95 (%)
YOLOv5n	**1,767,976**	68.16	75.60	47.60
YOLOv8n	3,006,233	71.67	79.62	50.75
YOLO11n	2,590,425	72.63	80.46	51.78
YOLOv12n	2,520,249	69.88	76.83	50.63
Ours	2,267,861	**75.22**	**82.92**	**54.13**

Table 5. Comparison of detection results of different algorithms on the CCTSDB2021 dataset.

Algorithm	Mandatory (%)	Prohibitory (%)	Warning (%)
YOLOv5n mAP@50	68.6	74.9	82.6
YOLOv8n mAP@50	73.2	77.6	86.4
YOLO11n mAP@50	74.7	81.2	84.6
YOLOv12n mAP@50	71.5	75.7	83.2
Ours mAP@50	**75.7**	**84.1**	**87.6**
YOLOv5n mAP@50-95	43.9	48.3	50.0
YOLOv8n mAP@50-95	48.7	50.5	53.1
YOLO11n mAP@50-95	49.4	53.7	52.3
YOLOv12n mAP@50-95	48.3	50.8	52.9
Ours mAP@50-95	**52.2**	**56.2**	**54.0**

To further evaluate the generalization ability of the proposed algorithm, additional experiments were conducted on the TT100K-2021 dataset, which encompasses a broad variety of traffic sign classes. Owing to its rich diversity, TT100K-2021 was used to train and validate the improved model. As shown in Table 6, the improved network demonstrated superior performance again, achieving the highest detection accuracy. Compared with YOLOv5n, mAP@50, mAP@50–95, and recall improved by 13.35%, 12.28%, and 9.48%, respectively.

Table 6. Comparison of detection results of different algorithms on the TT100K-2021 dataset.

Algorithm	Parameter Count	Recall (%)	mAP@50 (%)	mAP@50-95 (%)
YOLOv5n	1,767,976	66.51	71.79	53.20
YOLOv8n	3,006,233	73.82	82.05	62.23
YOLO11n	2,590,425	75.15	82.69	62.88
YOLOv12n	2,520,249	65.49	75.26	57.08
Ours	**2,267,861**	**75.99**	**85.14**	**65.48**

Moreover, we conducted comparative experiments of the network on the VisDrone2019 dataset to verify the generalization capability of the proposed algorithm. The VisDrone2019 dataset created by a team from Tianjin University; it contains millions of annotated objects and covers tasks such as object detection and tracking under complex unmanned aerial vehicle (UAV) viewpoints. As shown in Table 7, on the VisDrone2019 dataset, the proposed network achieves better detection performance while remaining lightweight. Compared with the YOLO11n baseline, the parameter count is reduced from 2,585,102 to 2,262,538 (a decrease of approximately 12.48%), while recall improves from 34.62% to 36.08%, mAP@50 increases from 34.84% to 35.94%, and mAP@50-95 rises from 20.31% to 21.08%. These results indicate that the proposed network provides superior accuracy and generalization under complex UAV viewpoints.

Table 7. Comparison of detection results of different algorithms on the Visdrone2019 dataset.

Algorithm	Parameter Count	Recall (%)	mAP@50 (%)	mAP@50-95 (%)
YOLO11n	2,585,102	34.62	34.84	20.31
Ours	**2,262,538**	**36.08**	**35.94**	**21.08**

4.4. Edge Device Deployment Experiments

To further assess the practicality of the proposed method on resource-constrained platforms, the model was deployed on an NVIDIA Jetson Orin NX 16 GB developer kit for evaluation. This device integrates an 8-core ARM CPU and an NVIDIA GPU based on the Ampere architecture, and runs NVIDIA Jetson Linux (L4T 36.4.7). During deployment, the model is first exported to the ONNX format and then optimized for inference using TensorRT. All experiments on the edge device use a single input stream with a resolution of 384×640 and a batch size of 1.

On the Jetson Orin NX, the baseline network achieves an end-to-end average inference latency of approximately 42 ms per frame (3.8 ms for preprocessing, 34.3 ms for network inference, and 3.8 ms for post-processing), corresponding to a real-time performance of about 24 FPS. In comparison, the proposed network attains an end-to-end average latency of around 32 ms per frame (4.0 ms for preprocessing, 23.6 ms for inference, and 4.4 ms for post-processing), corresponding to a real-time performance of about 31 FPS. During inference, the total power consumption of the Jetson Orin NX module is approximately 12.5 W under the baseline network load and about 11.5 W under the proposed network load. For both models, the memory usage is roughly 4 GB (out of 16 GB total), and the chip temperature remains below 50 °C, indicating substantial headroom in both computational and thermal budgets.

Compared with a desktop GPU platform, the inference speed on the Jetson Orin NX is reduced, while the detection accuracy (mAP) is essentially preserved. These results demonstrate that the proposed method can operate efficiently on low-power edge devices.

4.5. Visualization Analysis

For intuitively demonstrating the effectiveness of the improved algorithm, Figures 8 and 9 illustrate the comparative detection results of the improved network and the conventional YOLO11n algorithm under various scenarios. In complex scenarios, Figure 8a,c display the detection outcomes of YOLO11n, while Figure 8b,d present those of the improved network. Notably, the proposed network successfully detects significantly more small mandatory traffic signs in Figure 8b,d.



Figure 8. Comparison of tiny object detection across different algorithms: (**a**,**c**) YOLO11n algorithm; (**b**,**d**) improved algorithm.

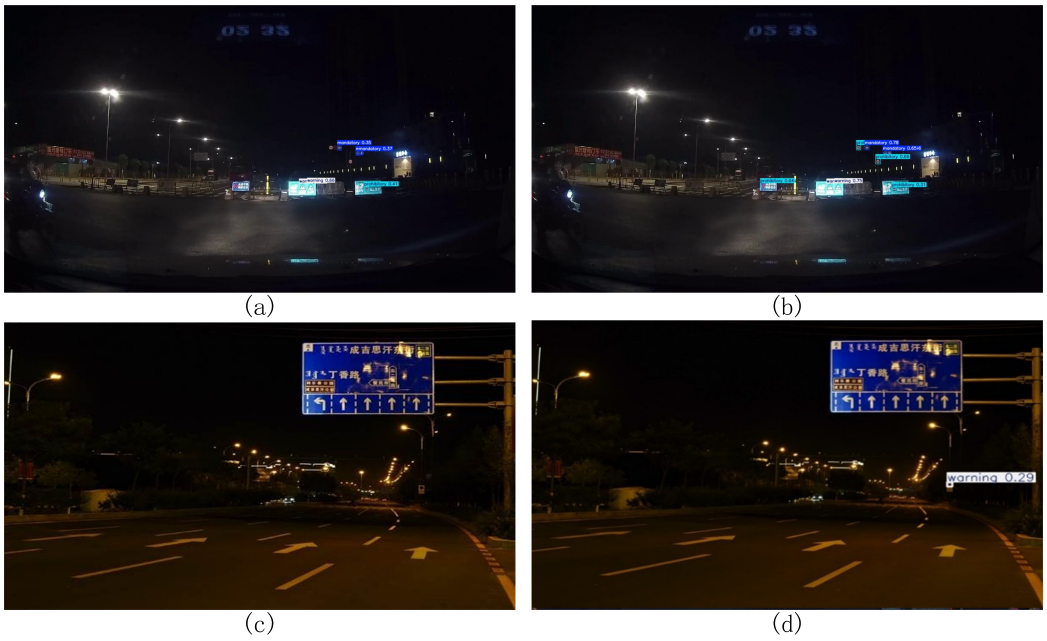

Figure 9. Comparison of detection under low-light conditions across different algorithms: (**a**,**c**) YOLO11n algorithm; (**b**,**d**) improved algorithm.

Similarly, under low-light conditions, Figure 9b identifies a more comprehensive set of traffic signs compared to Figure 9a,d and detects a warning sign that was undetected in

Figure 9c. The improved network integrates a multi-scale attention mechanism and feature enhancement modules, significantly improving its ability to represent low-contrast and tiny targets. These enhancements facilitate the extraction of richer multi-scale and multi-semantic features, boosting robustness against occlusion, deformation, and challenging lighting conditions.

To further evaluate the robustness of the algorithm from a quantitative perspective, we construct a more challenging subset based on the original validation set. Specifically, the RGB images are converted to the YCrCb color space, and samples are selected if they satisfy at least one of the following conditions: mean luminance meanY < 70, luminance standard deviation stdY < 35, or more than three target instances in a single image. We re-evaluate the conventional YOLO11n and the improved network on this subset and define a relative robustness metric (RR) to characterize the relative decrease in detection accuracy between the full validation set and the challenging subset: a smaller RR value (i.e., closer to 0) indicates better robustness. The RR metric is computed as

$$\mathrm{RR} = 1 - \frac{\mathrm{mAP}_{\mathrm{cor}}}{\mathrm{mAP}} \tag{14}$$

where mAP denotes the mean Average Precision on the full validation set, and $\mathrm{mAP}_{\mathrm{cor}}$ denotes the mean Average Precision on the challenging subset. The experimental results show that, on this challenging subset, the improved network consistently achieves a better RR (i.e., a smaller performance drop) than YOLO11n, further confirming its superior robustness under complex illumination and densely populated target scenarios.

As shown in Table 8, both algorithms exhibit a certain degree of performance degradation on the challenging subset; however, the proposed network consistently outperforms YOLO11n across all evaluation metrics. In particular, the recall of the proposed method increases from 59.10% to 64.80%, indicating that a larger proportion of targets can still be correctly detected under complex illumination and dense-object conditions. In terms of detection accuracy, the proposed network achieves 72.60% mAP@50 and 43.70% mAP@50–95, surpassing YOLO11n by 4.00% and 2.50%, respectively.

The relative robustness (RR) indices further substantiate the superiority of the proposed approach. For YOLO11n, RR(mAP@50) and RR(mAP@50–95) are 0.1463 and 0.2043, whereas the corresponding values for the proposed network are reduced to 0.1244 and 0.1921. Since a smaller RR value (i.e., closer to 0) corresponds to a smaller performance drop from the full validation set to the challenging subset, these results demonstrate that the proposed network not only yields higher detection accuracy, but also maintains stronger robustness when confronted with low-light, low-contrast, and densely distributed targets.

Table 8. Comparison of detection results on the challenging subset.

Algorithm	Recall (%)	mAP@50 (%)	mAP@50-95 (%)	RR(mAP@50)	RR(mAP@50-95)
YOLO11n	59.10	68.60	41.20	0.1463	0.2043
Ours	**64.80**	**72.60**	**43.70**	**0.1244**	**0.1921**

In summary, compared to YOLO11n, the improved network demonstrates greater accuracy in detecting small and densely distributed objects under varying lighting conditions. It effectively reduces overlooked detections and false positives, demonstrating superior robustness and resistance to environmental interference.

5. Conclusions

Based on the traditional YOLO11n algorithm, this study proposes a novel network structure. In the deeper layers of the network, standard convolutional operations are replaced with ADown modules, effectively reducing the model's complexity. Furthermore, a

micro-detection head is incorporated at the output stage, accompanied by a multi-scale convolutional attention mechanism specifically tailored for micro-, small, and medium detection heads. These enhancements collectively constitute the proposed network architecture.

The modified model was primarily trained and validated on the CCTSDB2021 traffic sign dataset to establish its core performance benchmarks. As a supplementary experiment, additional training and validation procedures were conducted on the TT100K-2021 dataset to further assess its generalization across diverse data distributions. Experimental results indicate that, compared to the traditional YOLO11n, the improved network achieves 82.92% mAP@50, 54.13% mAP@50–95, and 75.22% recall on the CCTSDB2021 dataset, with only 2.27 M parameters and an inference speed of 150.51 FPS. On the TT100K-2021 dataset, the model achieves 85.14% mAP@50, 65.48% mAP@50–95, and 75.99% recall.

These results demonstrate that the proposed network exhibits exceptional performance in detecting tiny, dense, and complex targets while maintaining a lightweight structure. Owing to its high accuracy, low parameter count, and real-time inference speed, the model is well-suited not only for on-board traffic sign detection in intelligent vehicles, but also for a wide range of resource-constrained applications, such as urban traffic monitoring, mobile robotics, UAV-based perception, and other embedded vision systems. This versatility highlights the practical utility of the proposed architecture in real-world intelligent transportation and edge-computing scenarios. Despite these advantages, the proposed model still has certain limitations. For instance, its performance under extremely adverse conditions—such as severe motion blur, heavy rain, or fog and ultra-low illumination—has not been exhaustively evaluated and may still be suboptimal. In addition, the current design is specifically tailored to traffic sign detection with a relatively limited category scale, and its scalability to more diverse object categories and more complex multi-task settings remains to be systematically investigated.

In future, we plan to extend the model to broader perception tasks and incorporate more advanced robustness-enhancing strategies (e.g., adverse-weather augmentation and domain adaptation), so as to further improve its applicability in more challenging real-world scenarios.

Author Contributions: Methodology, H.F. and J.M.; Investigation, H.F. and Z.G.; Supervision, P.Z. and W.Z.; Writing—original draft preparation, H.F. and Z.G.; Writing—review and editing, Y.C.; Resources, Y.C.; Visualization, C.L. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by the National Natural Science Foundation of China (Grant No. 52205138), the commissioned project from China Automotive Data Co., Ltd. (Tianjin) (Project No. 2QSJTI-2-2023-00424), and the Graduate Research and Innovation Project by CAUC (Project No. 2023YJSKC01005).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest. Authors Jiaxu Meng, Pengchao Zhao, and Wenchao Zhang were employed by China Automotive Technology & Research Center Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest. The authors declare that this study received funding from China Automotive Data Co., Ltd. (Tianjin). The funder had the following involvement with the study: [Methodology, J.M.; Supervision, P.Z. and W.Z.].

References

1. Bojarski, M.; Del Testa, D.; Dworakowski, D.; Firner, B.; Flepp, B.; Goyal, P.; Jackel, L.; Monfort, M.; Muller, U.; Zhang, J.; et al. End to End Learning for Self-Driving Cars. *arXiv* **2016**, arXiv:1604.07316. [CrossRef]
2. Grigorescu, S.; Trasnea, B.; Cocias, T.; Macesanu, G. A Survey of Deep Learning Techniques for Autonomous Driving. *J. Field Robot.* **2020**, *37*, 362–386. [CrossRef]
3. Yurtsever, E.; Lambert, J.; Carballo, A.; Takeda, K. A Survey of Autonomous Driving: Common Practices and Emerging Technologies. *IEEE Access* **2020**, *8*, 58443–58469. [CrossRef]
4. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
5. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Region-Based Convolutional Networks for Accurate Object Detection and Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 142–158. [CrossRef] [PubMed]
6. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
7. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
8. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
9. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767. [CrossRef]
10. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot Multibox Detector. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
11. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475.
12. Sohan, M.; Sai Ram, T.; Rami Reddy, C.V. A Review on YOLOv8 and Its Advancements. In Proceedings of the International Conference on Data Intelligence and Cognitive Informatics, Singapore, 18–20 November 2024; pp. 529–545.
13. Fang, S.; Chen, C.; Li, Z.; Zhou, M.; Wei, R. YOLO-ADual: A Lightweight Traffic Sign Detection Model for a Mobile Driving System. *World Electr. Veh. J.* **2024**, *15*, 323. [CrossRef]
14. Qiu, J.; Zhang, W.; Xu, S.; Zhou, H. DP-YOLO: A Lightweight Traffic Sign Detection Model for Small Object Detection. *Digit. Signal Process.* **2025**, *165*, 105311. [CrossRef]
15. Li, J.; Tang, H.; Li, X.; Dou, H.; Li, R. LEF-YOLO: A Lightweight Method for Intelligent Detection of Four Extreme Wildfires Based on the YOLO Framework. *Int. J. Wildland Fire* **2023**, *33*, WF23044. [CrossRef]
16. Tang, Q.; Su, C.; Tian, Y.; Zhao, S.; Yang, K.; Hao, W.; Feng, X.; Xie, M. YOLO-SS: Optimizing YOLO for Enhanced Small Object Detection in Remote Sensing Imagery. *J. Supercomput.* **2025**, *81*, 303. [CrossRef]
17. Li, R.; Chen, Y.; Wang, Y.; Sun, C. YOLO-TSF: A Small Traffic Sign Detection Algorithm for Foggy Road Scenes. *Electronics* **2024**, *13*, 3744. [CrossRef]
18. Zhao, L.; Wei, Z.; Li, Y.; Jin, J.; Li, X. SEDG-YOLOv5: A Lightweight Traffic Sign Detection Model Based on Knowledge Distillation. *Electronics* **2023**, *12*, 305. [CrossRef]
19. Wang, C.-Y.; Yeh, I.-H.; Liao, H.-Y.M. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. In Proceedings of the European Conference on Computer Vision, Milan, Italy, 29 September–4 October 2024; pp. 1–21.
20. Zhang, J.; Zou, X.; Kuang, L.; Wang, J.; Sherratt, R.S.; Yu, X. CCTSDB 2021: A More Comprehensive Traffic Sign Detection Benchmark. *Hum.-Centric Comput. Inf. Sci.* **2022**, *12*, 23.
21. Zhang, J.; Xie, Z.; Sun, J.; Zou, X.; Wang, J. A Cascaded R-CNN with Multiscale Attention and Imbalanced Samples for Traffic Sign Detection. *IEEE Access* **2020**, *8*, 29742–29754. [CrossRef]
22. Zhang, J.; Wang, W.; Lu, C.; Wang, J.; Sangaiah, A.K. Lightweight Deep Network for Traffic Sign Classification. *Ann. Telecommun.* **2020**, *75*, 369–379. [CrossRef]
23. Zhu, Z.; Liang, D.; Zhang, S.; Huang, X.; Li, B.; Hu, S. Traffic-Sign Detection and Classification in the Wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2110–2118.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.