PED-YOLO: Lightweight Framework for Foggy Target Detection

Zhehao Zhang, Tianwei Shi*, Jianbang Ding, Ruiqi Wang.

Abstract—**This paper proposes a lightweight framework named PED-YOLO to mitigate the limitations of low detection accuracy, complex network architecture, and excessive parameters in fog detection tasks. The framework integrates the PConv, EfficientNetV2, and ADown modules. In the neck of the YOLO11 architecture, the PConv and ADown modules are employed to replace the original C3K2 and Conv components. This substitution effectively reduces the number of training parameters and decreases the computational load. The backbone integrates EfficientNetV2, resulting in lower architectural complexity and improved computational efficiency. The ADown module is incorporated in place of the original downsampling component. Its introduction is enhanced for downsampling performance and the parameter count is further reduced. Compared with the original YOLO11 detection network, the proposed PED-YOLO detection framework has the mAP50 value, parameter number, and GFLOPs reduced by 76.7%, 23.05%, and 27.18%, respectively. Experimental results demonstrate that the PED-YOLO object detection model achieves significant improvements in detection accuracy. At the same time, it reduces the number of parameters, lowers computational complexity. These enhancements establish it as a highly efficient and reliable solution for object detection tasks in foggy environments.**

Index Terms—**PConv, EfficientNetV2, ADown, lightweight, YOLO11, foggy target detection**

I. Introduction

Target detection in foggy conditions is a crucial aspect of autonomous driving technology, contributing significantly to driving safety and enhancing the operational efficiency of intelligent transportation systems. In such challenging environments, autonomous vehicles are required to accurately identify objects, including roads, vehicles, and pedestrians, while making precise driving decisions based on their observations. However, foggy conditions can cause a significant degradation in image quality. The contours of targets become blurred, and the contrast is markedly reduced. These effects pose substantial challenges and difficulties for target detection.

Improving the accuracy of target detection in foggy weather is of great significance for unmanned driving technology. It is closely related to the precision and reliability of the vehicle's decision-making system and further influences driving safety. High target detection accuracy enables vehicles to promptly and precisely identify and avoid obstacles, accurately recognize traffic signs and signals, thereby effectively reducing the risk of traffic accidents and simultaneously enhancing traffic flow and operational efficiency.

High target detection accuracy enables vehicles to promptly and precisely identify and avoid obstacles. It also helps accurately recognize traffic signs and signals, thereby reducing the risk of traffic accidents and improving traffic flow and operational efficiency. On one hand, the computing power and storage resources of embedded devices are limited. Too complex detection models will lead to slow processing and cannot achieve real-time operation. They will affect the overall response speed and safety of the unmanned driving system. On the other hand, reducing detection accuracy in the process of model lightweighting must be avoided, as this would compromise the ability to meet the requirements of practical applications. Therefore, it is urgent to develop a lightweight object detection model that maintains high detection accuracy under foggy conditions and runs efficiently on embedded devices. This would promote the further development and broader adoption of unmanned driving technology.

To solve the above problems, this paper proposes a lightweight detection framework (PED-YOLO) that combines technologies such as PConv, EfficientNetV2 and ADown. Specifically, PConv and ADown are adopted in place of the original Conv and C3K2 modules in YOLO11. This realizes the lightweight operation of the model. In the backbone network, the lightweight EfficientNetV2 model is incorporated to further reduce the overall weight of the framework. In addition, the ADown module is adopted in place of the traditional down-sampling module to enhance the model's down-sampling capability. These improvements enable the PED-YOLO framework to effectively reduce model complexity and maintain high target detection accuracy in harsh environments such as fog. At the same time, the framework satisfies the constraints of embedded devices regarding model size and operational efficiency.

Manuscript received May 21, 2025; revised July 31, 2025

This work was supported by Liaoning Science and Technology Department's 'Jiebang Guashuai' (Unveiling the List and Appointing the Leader) Project (2024-241).

This work was supported by Liaoning Provincial Department of Education University Basic Scientific Research Business Fee Project (LJ242410146070).

Zhehao Zhang is a postgraduate student of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, 114051, China (phone:+86-138-0420-7271; email:421753234@qq.com).

Tianwei Shi* is an associate professor at the School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China (corresponding author to provide phone: +86-139-9805-3962; e-mail: tianweiabbcc@163.com).

Jianbang Ding is a postgraduate student of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, 114051, China (phone: +86-155-6627-3253; email: jianbang0219@163.com).

Ruiqi Wang is a postgraduate student of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, 114051, China (phone: +86-186-0787-3380; email:1046913352@qq.com).

II. RELATED WORK

In recent years, deep learning models have occupied a dominant position in the field of object detection, with convolutional neural networks playing a particularly prominent role, and numerous remarkable achievements have been reported [1][2]. Deep learning models in the field of label detection can be primarily divided into two categories. The first category consists of two-stage detection algorithms based on region proposals, with typical representatives including Region Convolutional Neural Network (R-CNN) [3], Fast R-CNN [4], and Faster R-CNN [5]. Nevertheless, although two-stage detection algorithms perform well in terms of accuracy, they have limitations in speed. The main drawback is the complexity and size of the network architecture. It involves a large number of parameters, resulting in a time-consuming recognition process that cannot meet the strict real-time performance requirements of foggy target recognition tasks.

The second category includes single-stage detection algorithms, such as You Only Look Once (YOLO) [6] and Single Shot Multi-Box Detector (SSD) [7], both of which extract features directly from the network architecture to predict the category and location of the object. By introducing a hybrid structure block that combines a multi-scale parallel large convolution kernel module with an enhanced parallel attention module, Lu et al. achieved good results on image dehazing tasks using MixDehazeNet[8]. Li et al. applied a technique based on convolutional neural networks (CNN) to develop the AOD-Net[9] model for image dehazing. In response to the challenge of foggy target detection, Zhong et al. proposed the DR-YOLO[10] model. DR-YOLO effectively addressed the problem of foggy target detection by integrating the atmospheric scattering model and co-occurrence relationship graph into the detector as prior knowledge. The proposed model not only improves object detection accuracy but also maintains good real-time performance. Akhmedov, F., et al. [11] combined the YOLOV10[12] model with a dehazing algorithm to create an improved method for ship fire detection. This method enhanced the accuracy and reliability of ship fire detection in complex marine environments. Through the introduction of a two-branch network architecture and an attention feature fusion module, Chu et al. proposed D-YOLO[13], which effectively combines image restoration and object detection tasks at the feature level. This approach significantly improves the accuracy and robustness of object detection under adverse weather conditions. Babu, K.R., et al. applied an unsupervised domain adaptation technique, called R-YOLO[14], to enhance object detection performance under adverse weather conditions. This method provides a safer and more reliable visual perception solution for fields such as autonomous driving and robotics.

In summary, deep learning methods have become the mainstream approaches for object detection tasks, including remote sensing object detection [15] and foggy object detection. With the development of deep learning technology, such methods have been widely applied in foggy scenes. They rely on models like Convolutional Neural Networks (CNN) to automatically learn feature representations from foggy images, enabling object detection and recognition. For example, the YOLO series has become a popular choice for foggy object detection due to its speed and high accuracy. However, the current YOLO11 still shows some limitations. On one hand, the model's feature extraction ability requires further improvement. It is evident during detection of small and occluded targets. This often leads to missed detections and false detections. On the other hand, the model has high computational complexity and cannot easily meet real-time requirements in practical applications. At the same time, EfficientNetV2[16] also presents limitations. Although it offers advantages in parameter count and computation, its performance in object detection remains insufficient. For example, EfficientNetV2 may fail to effectively extract features during the detection of small-sized defect targets. This results in insufficient detection accuracy. The PConv[17] is mainly aimed at general convolutional neural network acceleration, but its advantages cannot be fully utilized in some specific tasks. Just like in some tasks that require highly customized convolution operations, the PConv may not be as effective as manipulation modules specifically designed for these tasks.

This paper focuses on detecting small and occluded targets in foggy environments by combining PConv, EfficientNetV2, and ADown modules within a lightweight design framework. Specifically, the PConv module reduces computational complexity and ensures the extraction of key features by optimizing traditional convolution operations. The ADown module addresses information loss during down-sampling and enables full fusion of multi-scale features. The backbone network using EfficientNetV2 significantly reduces parameter count and computation. It also improves the ability to capture subtle defect targets while preserving accurate feature extraction. The collaborative design of each module lowers the network's computational burden and maintains high detection accuracy and real-time performance. Quantitative experiments and feature visualization further confirm the effectiveness and advantages of this design in foggy target detection tasks.

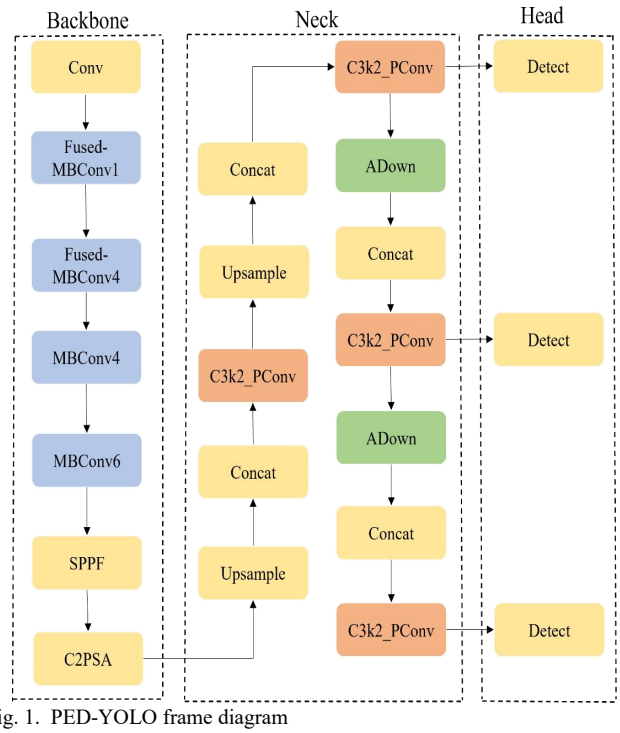


Fig. 1. PED-YOLO frame diagram

To tackle the challenges in foggy object detection, this paper proposes a lightweight detection framework named PED-YOLO, as shown in Fig. 1. It integrates the PConv module, the EfficientNetV2 module, and the ADown module for construction. In this framework, the network structure of PConv + ADown replaces the C3K2 and Conv modules in the original YOLO11 neck network. This modification reduces the number of parameters and computational load during training, thereby improving detection accuracy and achieving the goal of lightweight design. In the backbone of the detection model, the lightweight EfficientNetV2 is integrated to simplify the overall complexity and make the backbone more compact. In the neck network, the ADown module replaces the original down-sampling module. This replacement enhances the model's down-sampling performance and reduces the number of parameters in the detection model.

The core contributions of this paper can be summarised as follows:

1. The Conv and C3K2 modules in the original YOLO11 neck network are substituted by the ADown and PConv network structures.
2. The EfficientNetV2 module is incorporated into the detection model's backbone to simplify the network architecture and create a lightweight backbone.
3. The traditional downsampling modules in the neck network are swapped with the ADown module to enhance downsampling capabilities.

III. RESEARCH METHODOLOGY

This paper proposes an efficient object detection framework, PED-YOLO, as shown in Fig. 2. The framework utilizes EfficientNetV2 as the backbone network and applies its compound scaling strategy to build a multi-scale feature pyramid. This approach improves both the efficiency and accuracy of feature extraction. Firstly, in the backbone network of the detection framework, the four detection modules Fused MBConv1, Fused MBConv4, MBConv4 and MBConv6 in EfficientNetV2 are used instead of the C3K2 and Conv modules in YOLO11. This improves the feature fusion ability and computational efficiency of the model. Secondly, in the neck structure, PConv oriented toward local perception is innovatively introduced to substitute the standard convolution operation. This enhances the expression ability of local features, reduces computational complexity, and retains high-frequency detail information. Finally, a progressive downsampling module based on the ADown operator was designed. This module effectively reduces the amount of computation by gradually lowering the resolution of the feature map while preserving the integrity of key features. Through the above improvements, PED-YOLO simplifies the network structure and reduces computational complexity as well as model size. At the same time, detection accuracy is maintained, thus enhancing overall detection performance and efficiency.

A. PConv

As shown in Fig. 3, the PConv is a novel convolutional architecture designed to address the challenge of floating-point operations per second (FLOPS) in deep convolutional networks[18]. The core of the PConv is that it selectively convoluts some channels of the input data while leaving the other channels unchanged. By setting a partial rate, the proportion of channels participating in the convolution is flexibly controlled. This method reduces computational overhead and enhances feature extraction efficiency. Specifically, the PConv significantly lowers floating-point operations (FLOPs) and memory access frequency while preserving model accuracy.

The design philosophy of the PConv maximizes the utilization of information from all channels while minimizing computing resource consumption. In network architectures with multi-layer convolutions, high similarity between channels often leads to functional duplication. The PConv reduces this redundancy by convolving only a selected subset of channels, improving spatial feature extraction efficiency. Unlike traditional full-channel convolution, the PConv focuses on a portion of the input channels, reducing computational complexity and redundancy. The numerical description and computational analysis are detailed in Equation (1):

$$FLOPS_{Conv} = h \times w \times k^2 \times c^2 \tag{1}$$

where, h and w represent the height and width of the feature map, k and c represent the filter size, and the number of channels. However, in the PConv, the computational cost is as follows:

$$FLOPS_{PConv} = h \times w \times k^2 \times c_p{}^2 \tag{2}$$

where, c_p represents the number of channels participating in the partial convolution. Define the partial rate as $r = C / C_P$. For example, during $r = \frac{1}{4}$, one quarter of the channels participate in the convolution operation.

The PConv reduces computational cost and floating-point operations by utilizing the similarity between channel features and the redundancy in the feature map [19]. This approach is important for lightweight and efficient object detection models, especially in resource-limited scenarios such as UAVs and similar applications. In YOLO11, the C3K2 module adopts the CSPBottleneck structure. It applies two parallel convolution layers to improve both feature extraction speed and efficiency. During use of C3K2_PConv in lieu of C3K2, the PConv mechanism lowers computation and memory access by convolving only part of the channels. This leads to improved speed and accuracy. C3K2_PConv also supports integration with other methods, such as deformable convolution, to strengthen feature extraction for objects with different shapes and scales. By setting the partial rate, C3K2_PConv controls the number of channels involved in convolution. This reduces floating-point operations and memory access, making it more suitable for limited-resource environments. During training, it learns channel feature relationships efficiently and enhances detection performance in inference.

B. EfficientNetV2

This study adopts EfficientNetV2[20] as the backbone for YOLO11, providing high-precision feature extraction at low computational cost. The EfficientNetV2 backbone uses Fuded-MBConv1, Fuded-MBConv4, MBConv4, and MBConv6 modules to replace YOLO11's C3K2 and Conv

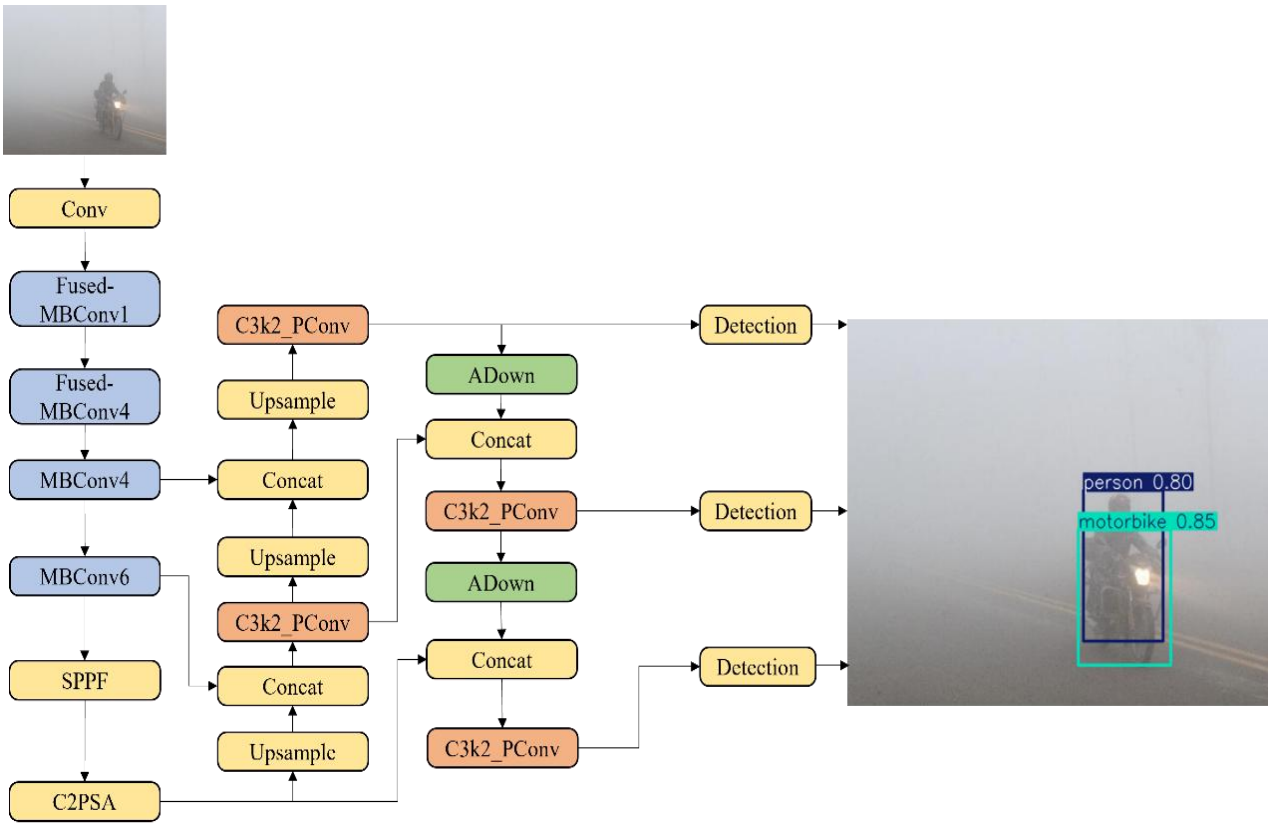


Fig. 2 Architecture of PED-YOLO

modules. EfficientNetV2 introduces the Fuded-MBConv module, offering a lightweight design and enhanced multi-scale feature extraction.

(1)Innovative module construction: Fused-MBConv module.

As shown in Fig. 4, one core innovation of EfficientNetV2 is the Fused Mobile Inverted Residual Bottleneck (Fused MBConv) module. This module alleviates redundant computation and repeated gradient propagation in traditional convolutional neural networks through structural optimization. Although the traditional depthwise separable convolution can reduce the amount of computation, its separate design (depthwise convolution + pointwise convolution) may lead to insufficient local information fusion in the feature extraction process, especially when dealing with high-resolution images, the laye-by-layer separation operation will introduce additional computational overhead.

Fused MBConv achieves optimization of the computation path by applying standard convolution to part of the deep convolution operations. This architecture both retains the advantages of being lightweight and strengthens the aggregation ability of local features. It thus serves to reduce the number of parameters (e.g., reducing FLOPs by about 30%) and improve the model's sensitivity to key features (e.g., edges and textures). In object detection tasks, this module can rapidly focus on foreground objects and suppress background interference by adjusting the receptive field of the feature map.For example, in intelligent traffic scenes involving dense vehicles, pedestrians, and changing illumination, Fused MBConv effectively extracts geometric and semantic features of targets. This provides low-latency support for real-time detection. Yin et al. [21] pointed out that using a progressive stacking strategy of modules, such as applying Fused MBConv in shallow layers and MBConv in deeper layers, helps prevent information loss in early stages caused by overly lightweight designs. This improves the balance between detection speed and accuracy.

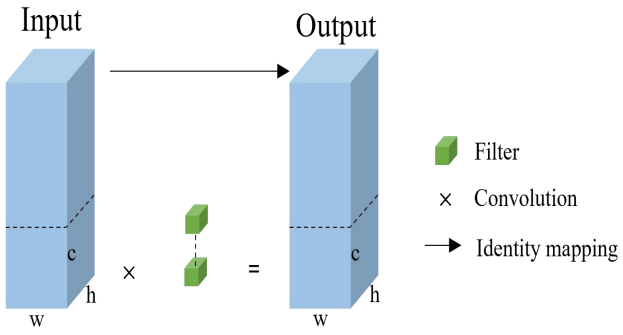


Fig. 3 PConv structure diagram

(2) Excellent lightweight design: parameter optimization and efficient resource utilization.

Through innovative network architecture design and parameter optimization, EfficientNetV2 reduces the number of model parameters to nearly one-tenth the scale of traditional backbone networks. At the same time, it maintains high detection accuracy. This lightweight characteristic lowers the demand for computing resources and supports deployment on edge computing devices and mobile platforms.The YOLO11 model built on this architecture achieves a good balance between detection accuracy and operational efficiency in monitoring scenarios. Consequently, the proposed model demonstrates strong potential for deployment in resource-constrained and real-time intelligent sensing scenarios.

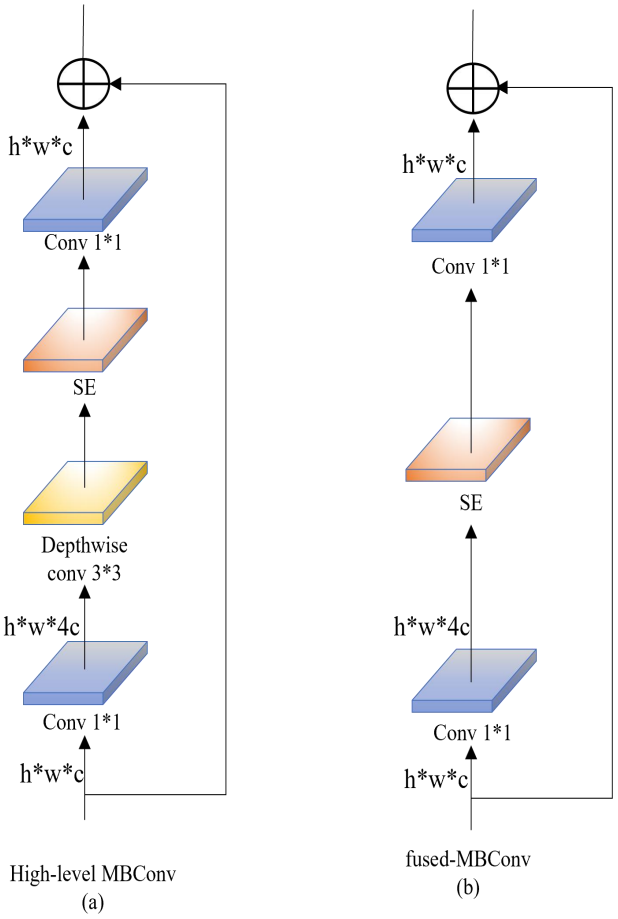


Fig. 4 High-level (a) MBConv and (b) Fusion-MBConv structures

(3) Enhancement of Multi-Scale Feature Extraction: Synergy Between ASPP and SE-Net.

(a) ASPP Module for Multi-Scale Feature Extraction:

The ASPP module integrated with EfficientNetV2 provides strong support for multi-scale feature extraction. In the complex actual object detection scene, the scale of the target object varies widely. These objects range from small microscopic parts and insects to large buildings and vehicles. Traditional backbone networks often struggle to extract feature information from objects with different scales in a fully accurate manner. By using 3×3 dilated convolution kernels with different dilation factors (e.g., 6, 12, 18), the ASPP module can sample and analyze the features of the input image from multiple scales, so as to obtain rich and comprehensive feature expressions. For small targets, the small expansion rate convolution kernel was used to fine extract details. For large objects, the large expansion rate convolution kernel is used to obtain global features. In chip defect detection within the electronic chip manufacturing industry, circuit and component sizes are small. The detection process requires extremely high accuracy. The ASPP module enables the YOLO11 model, based on EfficientNetV2, to capture key features of subtle defects on the chip. It prevents missed detections caused by very small targets. This enhances both the accuracy and reliability of chip defect detection and provides strong technical support for ensuring the quality of electronic products.

(b) SE-Net Attention Mechanism:

In the field of object detection, the problem of sample imbalance is particularly prominent. Especially in the one-stage detector, the number of background samples (negative samples) far exceeds that of foreground samples (positive samples). This causes the model to focus on learning a large number of easily classified background samples and ignore the learning of a small number of important foreground samples. To solve this issue, a novel sample equalization strategy, SE, is proposed. The SE block consists of two steps: extrusion and excitation. In the extrusion step, global average pooling is applied to each channel to generate channel descriptors. In the excitation step, channel weights are learned through two fully connected layers and a Sigmoid activation function. These weights are then applied to the original feature map to produce the final output [22]. The main idea is to redefine the loss function so that the model treats each class more fairly during training. Specifically, a dynamic adjustment mechanism is introduced to balance the impact of samples with different difficulty levels on the total loss. This improves the overall performance of the model.

Mathematical statement: Let P be the probability predicted by the model and Y represent the actual label.

For the binary classification problem, the basic cross-entropy loss is:

$$L_{CE}(P,Y) = -Ylog(P) - (1-Y)log(1-P) \tag{3}$$

To deal with the problem of sample imbalance, an improved loss function is proposed. It combines the sample difficulty adaptive factor and the class weight adjustment factor. The form is as follows:

Improved Loss Function:

$$L_{SE}(P,Y) = -W(Y)\cdot[Ylog(\sigma(P)) + (1-Y)log(1-\sigma(P))] \tag{4}$$

where $\sigma(P)$ is the Sigmoid function, which maps the original probability value to the interval (0, 1), $W(Y)$ is a weighting factor calculated based on the proportion of sample classes to enhance the influence of minority classes.

In addition, to further emphasize the importance of complex samples, a regularisation factor γ based on sample prediction error is introduced:

Regularisation Factors:

$$\gamma(P,Y) = |\ Y - P|^{\delta} \tag{5}$$

where, δ is a hyperparameter that controls the strength of the regularisation factor. The final SE loss function can be expressed as:

Final Loss Function:

$$L_{final}(P,Y) = \gamma(P,Y)\cdot L_{SE}(P,Y) \tag{6}$$

Applying the SE method, the model can more effectively learn key foreground information. It also reduces the negative impact of sample imbalance, leading to significant improvements in accuracy and recall rate for the object detection task.

(4) EfficientNetV2 backbone network adaptation.

In this work, the architecture of YOLO11 is improved by replacing its default backbone network with EfficientNetV2. The key implementation steps are as follows:

1) Input layer adaptation: The input resolution is adjusted from the original 640×640 to 480×640, in line with the recommended input ratio of EfficientNetV2. Bilinear interpolation is used to maintain the spatial continuity of the feature map.

2) Feature extraction layer migration: The outputs from Stage1 to Stage7 of EfficientNetV2 (stride=4 to stride=32)

are intercepted and used in place of the CSPDarknet53 module in the original backbone network. To resolve the channel dimension mismatch, a 1×1 convolutional regulator (with a channel number of 512→256) is added before the neck network. Layer normalization is then applied to stabilize the feature distribution.

3) Parameter optimisation strategy:

(a) Freeze fine-tuning: The parameters of the first 15 layers of EfficientNetV2 (about 68% of the total number of parameters) are frozen. Only the parameters in Stage6, Stage7, and the subsequent detection heads are fine-tuned.

(b) Calculation compression: The 3×3 standard convolution in the SPPF module of YOLO11 is replaced by a depthwise separable convolution, thereby reducing the computational complexity of module k.

C. ADown

As shown in Fig. 5, the ADown module is an improved convolution module. It is designed to solve the problem of losing detailed information caused by the traditional downsampling method in object detection tasks. Through a series of carefully designed operations, the module can reduce the size of the feature map while retaining more detail information, and enhance the ability of the model to capture features at different scales. The ADown module functions as a substitute for the convolution module in the backbone network. As a result, the model captures finer image details and lowers both the computational load and structural complexity [23].

(1) The core of the ADown module is its unique combination of pooling and convolution operations:

1) Average pooling: It is performed on the input feature map $F \in R^{H\times W\times C}$ and halving its size.

$$F_{avg} = AvgPool(F) \tag{7}$$

where, $AvgPool(\cdot)$ denotes an average pooling operation that reduces the spatial dimensions of the input feature map from $H\times W$ to $\frac{H}{2}\times\frac{W}{2}$.

2) segmentation process

The average-pooled feature map, F_{avg} , is divided into two parts, x_1 and x_2 , along the channel dimension.

$$x_1, x_2 = Split(F_{avg}) \tag{8}$$

3) Local feature extraction

For the first partitioned part, x_1 , a 3×3 convolutional layer (stride 2, padding size 1) is used to extract local features.

$$x_1' = Conv1(x_1; w_1, b_1) \tag{9}$$

where w_1 and b_1 represent the weights and biases of the convolutional kernel, respectively.

4) Global feature extraction

For the second part, x_2 , a maximum pooling operation is first performed, followed by a 1×1 convolutional layer (stride 1, padding 0) to capture more abstract global features.

$$x_2' = Conv2(MaxPool(x_2); W_2, b_2) \tag{10}$$

5) Feature stitching

Finally, the feature maps from the two components are concatenated along the channel dimension. This generates the final output.

$$F' = Concat(x_1', x_2') \tag{11}$$

where, $Concat(\cdot)$ denotes the operation of concatenating two feature maps along the channel dimension.

(2) Neck network optimization based on ADown

To solve the problem of computational redundancy in the neck network of YOLO11, this study proposes using the ADown downsampling module instead of the original 3×3 standard convolution downsampling module. Specifically, the feature map downsampling operation that uses a convolution kernel with stride 2 in the original model is substituted by the cascaded structure of the ADown module. Firstly, the global information of the feature map is retained by average pooling. Then the channel dimension is compressed using 1×1 convolution. Finally, the spatial dimension undergoes fine-grained feature extraction through 3×3 depthwise separable convolution.

This improvement both significantly reduces the computational complexity and effectively alleviates the loss of high-frequency information during the traditional downsampling process. It enhances the ability of multi-scale feature fusion. Experiments show that the substitution strategy improves the representation ability of the neck network for small target features by about 3.2%, and the number of parameters remains basically unchanged.

In summary, the ADown module significantly improves the performance of object detection models in remote sensing images and other complex scenes. It achieves this through unique pooling, convolution, and feature concatenation operations. The module both retains more detailed information during the downsampling process and effectively captures the key features of the target. This leads to an overall improvement in target detection performance. The ADown module reshapes the down-sampling operation. This change reduces the model's computational load and enhances its ability to retain target features [24].

IV. Experiments and Analysis

A. Experimental Settings

The improved YOLO11 detector is chosen as the core network. The input image is uniformly resized to 640×640. During training, the initial learning rate is set to 0.01, the momentum is set to 0.9, and the weight decay parameter is 0.0005. The number of epochs and batch size are set to 400 and 8, respectively. The experiments are conducted using the PyTorch framework. All training and testing procedures are carried out in a Windows 10 environment with CUDA 11.2 support. The hardware platform consists of an Intel® Xeon® Silver 4214 processor, an NVIDIA RTX 3080Ti GPU with 12GB of video memory, and 64GB of system memory. This configuration ensures stable and efficient model training, and the network achieves reliable convergence under the above settings, allowing the model to perform to its full potential. The detection performance is evaluated using mean Average Precision (mAP) as the primary metric.

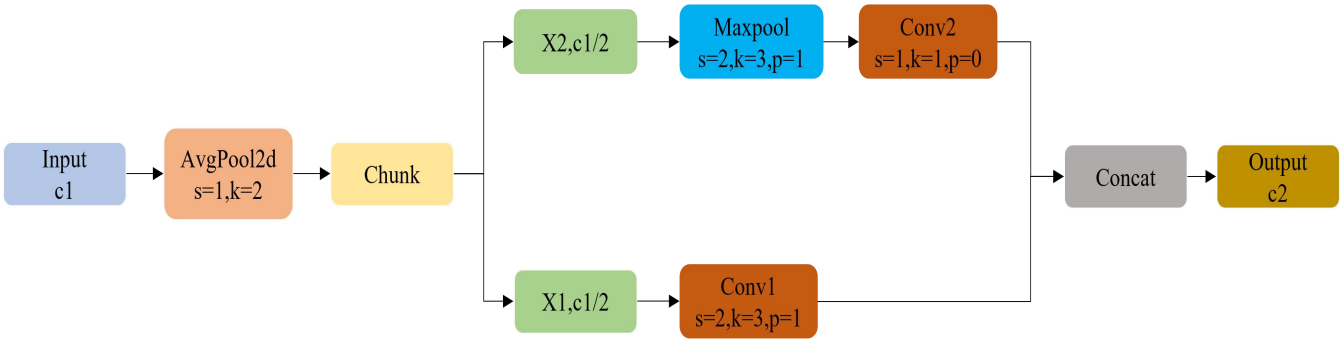


Fig. 5 ADown Convolution module

Fig. 6 RTTS Dataset

B. Datasets

As shown in Fig. 6, the experiment uses the RTTS dataset to comprehensively evaluate the improved YOLO11 detection network. The dataset was carefully collected and constructed by the Computer Vision research group at Saarbrueken University, Germany, to support research in real-time traffic sign detection. It is a large-scale, high-resolution real-time object detection set that contains rich image resources. This dataset offers valuable data for the research and development of object detection algorithms across various environments and conditions. RTTS is a challenging test set designed to evaluate the adaptability and performance of object detection algorithms under hazy conditions. It includes diverse scenes and object categories and provides accurate annotation information, making it an important benchmark for research in domain adaptation for object detection algorithms [25]. The RTTS dataset is also a real-world, task-driven test set for evaluating object detection algorithms in foggy conditions. It features diverse images of foggy scenes and labels objects in different categories and locations. These characteristics provide rich experimental data for studying foggy object detection [26].

C. Evaluation index

1) mAP (mean Average Precision) is a metric used to evaluate performance in object detection tasks. It takes into account the differences between target classes. To calculate mAP, the Average Precision (AP) is first computed for each class. Then, the average of the AP values across all classes is determined. mAP@50 refers to the mean Average Precision when the IoU (Intersection over Union) threshold is set to 0.5.

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{12}$$

where N represents the number of categories.

2) Precision (P) refers to the proportion of actual positive samples. These samples are among all samples predicted as positive by the model. It measures the accuracy of the model predictions, defined as the proportion of correct identifications in the prediction results, expressed mathematically as

$$P = \frac{TP}{TP + FP} \tag{13}$$

where TP is the number of true positives, and FP is the number of false positives.

3) Recall (R) is the proportion of actual positive samples identified by the model. These samples are among all positive samples. It quantifies the model's ability to detect all relevant positive instances. Mathematically expressed as

$$R = \frac{TP}{TP + FN} \tag{14}$$

where FN is the number of false negatives.

4) FLOPs (Floating-Point Operations) quantify the total computational workload required by a model during inference and are typically expressed in GFLOPs.

$$FLOP_s = C_{in} \times K^2 \times H \times W + C_{in} \times C_{out} \times H \times W \tag{15}$$

these symbols are defined as follows: C_{in} is the number of input channels, C_{out} denotes the number of output channels, K is the size of the convolution kernel, and H , W represents the height and width of the output feature map.

D. Contrast Experiments

In the experimental section, seven different technical routes are included in the comparison category. These routes are YOLOV5s , YOLOV7-tiny ,YOLOV8 [27], YOLOV10, YOLO11, YOLOV12,and the method proposed in this study. Experiments are conducted on the RTTS dataset. During the experiment, all methods adopt the same training strategy and hyperparameter settings. This ensures the fairness and comparability of the comparison results.

As shown in TABLE I below, based on the comparison results of various key indicators, the method proposed in this study demonstrates significant advantages in model size, parameter number, FLOPs, accuracy, recall rate, and mAP50. As shown in TABLE II , the model achieves a high level of

TABLE I
COMPARATIVE EXPERIMENTAL RESULTS OF EACH MODEL IN THE RTTS DATASET

MODELS	Size	Parameter	FlOPs	P	R	mAP50
YOLOV5s	13.7	26.9	15.8	87	48.9	68
YOLOV7-tiny	6.7	36.9	13.3	75.5	58.3	72.16
YOLOV8	5.98	2.87	8.2	76.5	71.3	76
YOLOV10	5.54	2.59	8.4	79.7	67.4	75.7
YOLO11	5.25	2.47	6.4	74.9	72.1	76
YOLOV12	5.21	2.51	5.8	77.8	65.5	74.6
Ours	**4.04**	**1.8**	**4.8**	**78.7**	**68.1**	**76.7**

TABLE II
COMPARISON OF THE PERFORMANCE OF EACH MODEL IN THE RTTS DATASET

Models	AP/%					mAP50
	Bicycle	Bus	Car	Motorbike	Person	
YOLOV5s	60.8	59.2	76.4	47.7	46.4	68
YOLOV7-tiny	62.9	59.5	76.3	50.2	49.3	72.16
YOLOV8	68.9	62.3	89.5	75.6	85.5	76
YOLOV10	69.5	57.5	88.2	79.3	84.1	75.7
YOLO11	67.8	63.9	89	74.3	84.8	76
YOLOV12	66.9	55	84.5	71.4	80.2	74.6
Ours	**70.2**	**68.9**	**88.4**	**80.9**	**85.3**	**76.7**

AP value and mAP50 in categories such as Bicycle, Bus, Motorbike, Person, and Car. The overall comprehensive performance remains the best. This result strongly proves that the performance of the proposed method in this study exceeds other compared methods in the object detection task. In particular, it is worth mentioning that the method of this study successfully achieves a comprehensive reduction of key indicators such as model size, parameter quantity, and model complexity. At the same time, it maintains a stable detection accuracy and effectively improves the lightweight degree of the model.

E. Ablation experiments

To verify the effectiveness of the improved method in this paper, comparative experiments are conducted under the same training strategy and hyperparameters. The effectiveness and feasibility of the improved method are confirmed by comparing the experimental results. As shown in TABLE III below, with the gradual introduction of PConv, EfficientNetV2 and ADown modules, the model size, parameter number and FLOPs are significantly reduced. At the same time, high detection accuracy is maintained. This achieves the lightweight optimization of the model. Specifically, the model size, number of parameters and FLOPs are significantly reduced during sole use of the PConv module. When PConv and EfficientNetV2 are both applied, the number of parameters decreases substantially and the model becomes more lightweight. During application of PConv, EfficientNetV2 and ADown, the final model size is reduced by 23.05% compared to the original YOLO11 model. The number of parameters decreases by about 27.18%, FLOPs are reduced by about 25.00%, and mAP50 increases by about 0.92%. These data show that by gradually introducing these modules, the model achieves significant results in terms of lightweight design. At the same time, the detection performance also improves.

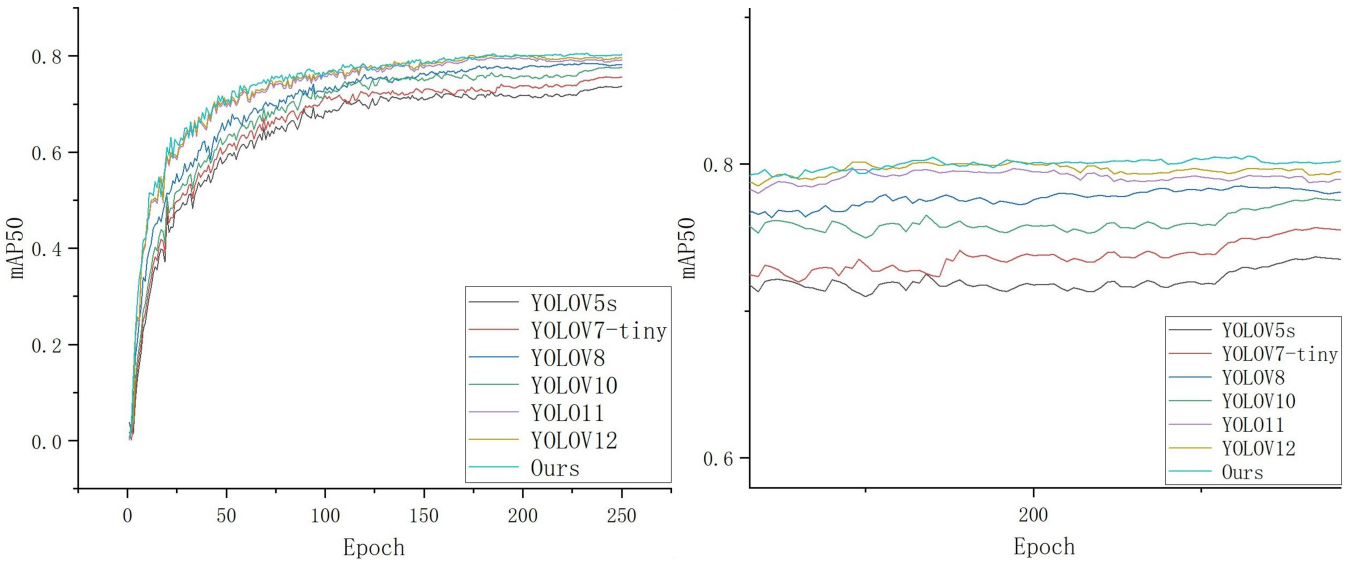


Fig. 7 mAP50 visual comparison diagram

TABLE III
RESULTS OF ABLATION EXPERIMENTS ON RTTS DATASETS

YOLO11	PConv	EfficientNetV2	ADown	Size	Parameter	FLOPs	mAP50
√				5.25	2590815	6.4	76
√	√			5.09	2507535	5.9	76.1
√	√	√		4.29	2019707	5.0	76.4
√	√	√	√	4.04	1886587	4.8	76.7

Fig. 8 Comparison of test results

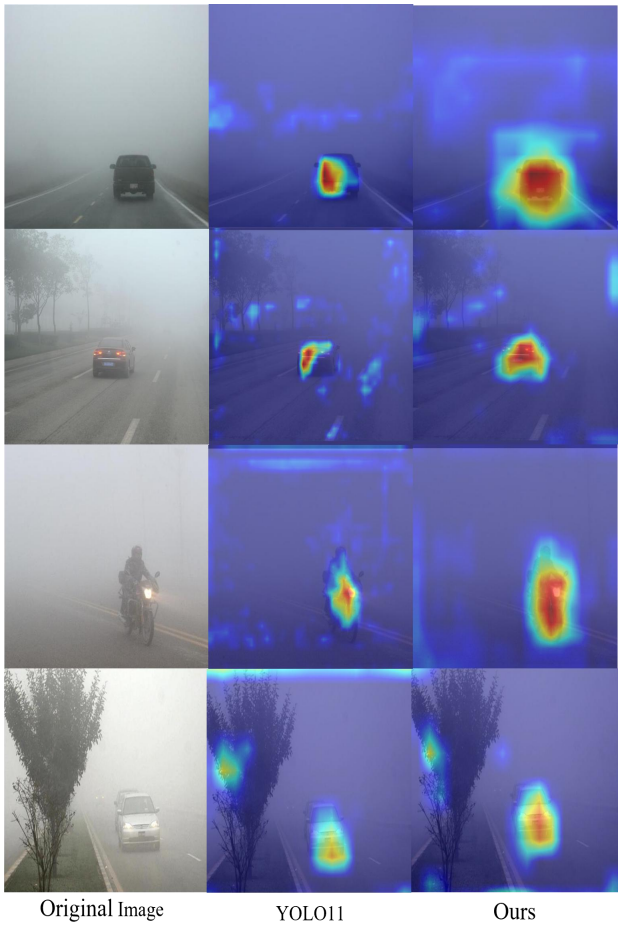


Fig. 9 Heatmap Comparison

F. Visualization-Based Experimental Analysis

The Fig.8 presents a detailed visual analysis of the detection results on the RTTS dataset, aimed at verifying the effectiveness of the proposed method in the fog detection task. The results are comprehensively compared with the original YOLO11 model. The comparative analysis clearly demonstrates that the proposed method outperforms the original YOLO11 model in terms of detection accuracy. The proposed method achieves higher detection accuracy on the RTTS dataset, accurately identifying and locating the target object in foggy environments, particularly in complex scenes and low visibility conditions. This result not only highlights the advantage of the proposed method in enhancing detection performance but also shows its robustness and adaptability for practical applications.

The Fig.9 illustrates the use of the Grad-CAM heatmap method to generate a color depth map, reflecting the weight of the detection classification in the identified area. In the original YOLO11 model, the heatmap does not focus on the detected area, and the weight of the detection classification is relatively small. In comparison, the heatmap generated by the proposed method shows a darker color in the detected area, indicating a higher weight for the detection classification. This improves the model's detection performance.

G. Experimental Analysis Using Precision-Recall Curves and Confusion Matrices

The PR curve [28] is a key tool for evaluating model performance in object detection and machine learning, especially when dealing with imbalanced datasets. After training on the RTTS dataset, the performance curve was plotted using precision (P) and recall (R) metrics. The ordinate represents precision, and the abscissa represents recall. As shown in FIG. 10, the curve for the proposed method is closer to the upper-right region of the coordinate axis. This indicates that, compared to the original YOLO11 model, the proposed method offers significant advantages in detection performance. It achieves higher accuracy and maintains a higher recall rate, which enables more precise identification of positive samples. This highlights not only the model's efficiency in detection tasks but also its superior performance in complex scenes. To better reflect the model's performance, the outputs of YOLO11 and PED-YOLO are compared. As shown in Figure 11, in the accuracy-confidence curve, the average accuracy of

PED-YOLO (blue line) is higher than that of YOLO11 across different confidence levels. In addition, in the recall-confidence curve, PED-YOLO outperforms YOLO11 in recall at various confidence levels. This demonstrates that our model significantly improves both precision and recall, enhancing its ability to identify objects in fog more accurately.

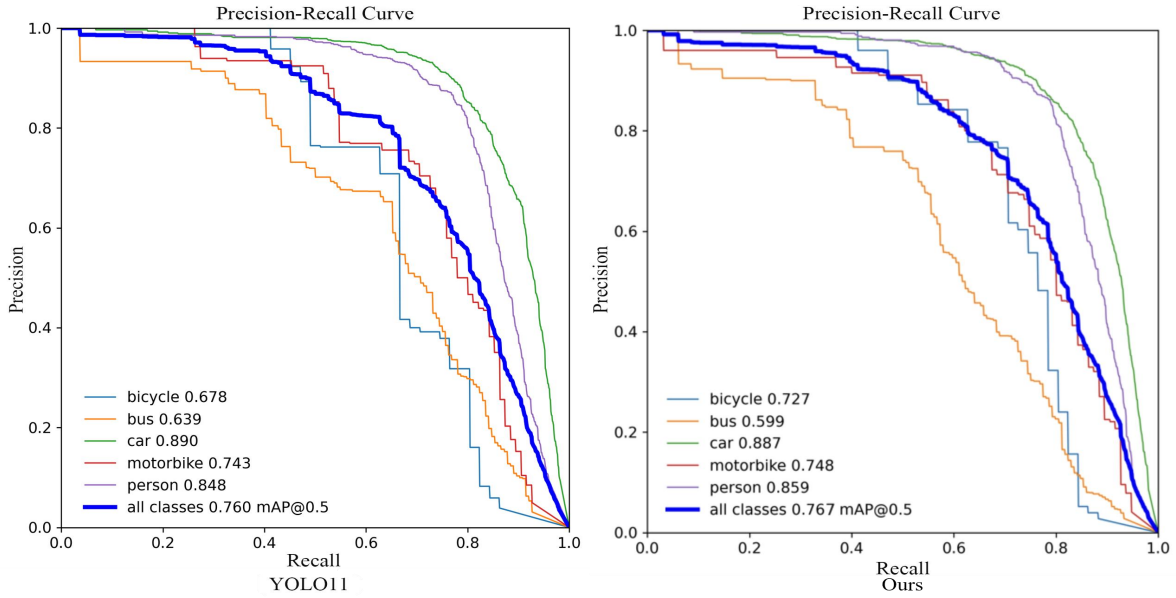
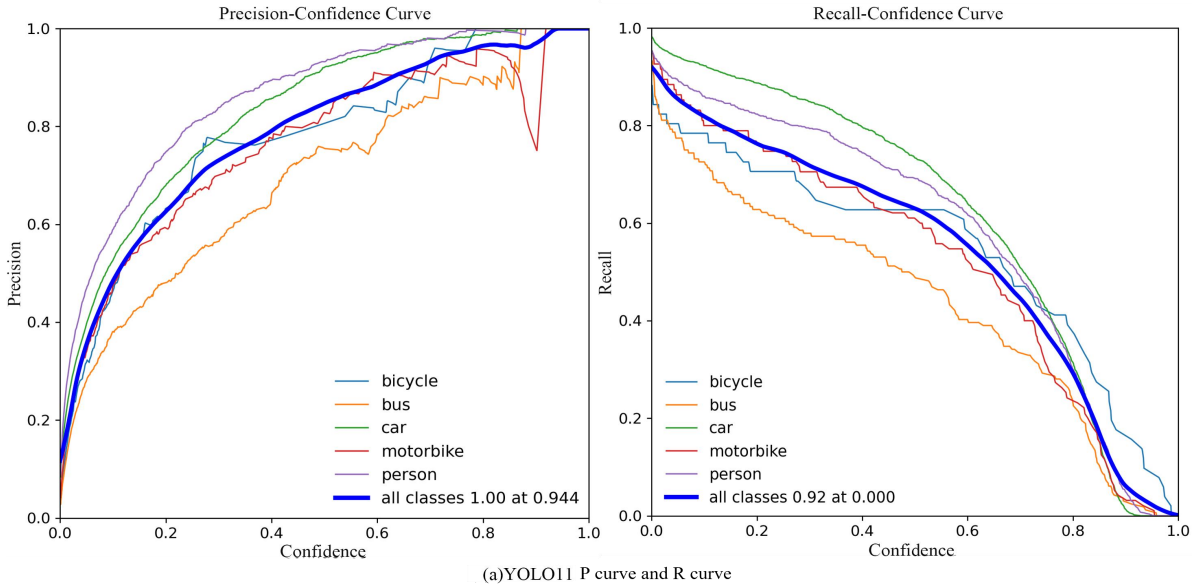
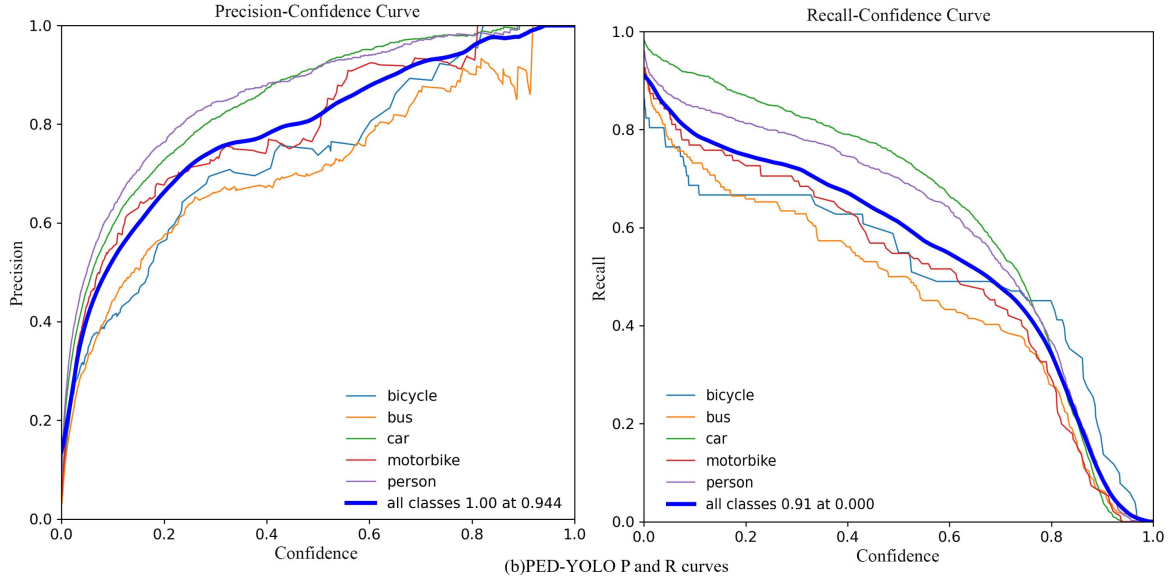


Fig. 10 PR Curve



(a)YOLO11 P curve and R curve



(b)PED-YOLO P and R curves

Fig. 11 Comparison of accuracy and recall rate before and after improvement

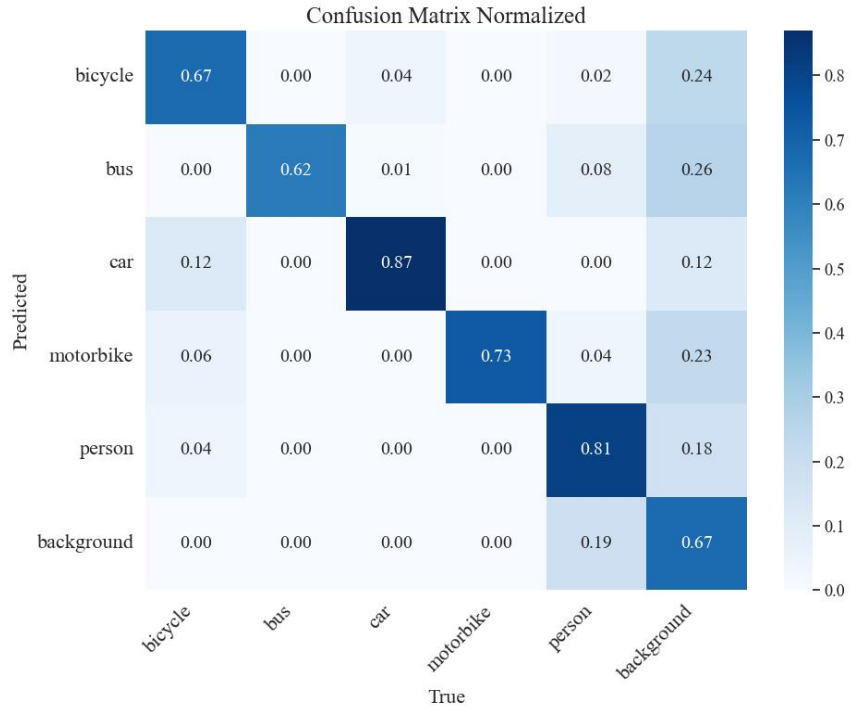


Fig. 12 Normalized confusion matrix

After normalization, the standard confusion matrix converts the values into proportions or percentages to eliminate the influence of class sample size differences. This makes the comparison of classification performance more intuitive. This method offers a fairer evaluation of model performance, particularly when the classes are unevenly distributed.

To evaluate the improved method, the normalized confusion matrix was used. The rows of the matrix represent the true class, and the columns represent the predicted class. The diagonal values indicate correct classification, while the bottom left and top right corners represent missed and false detections, respectively. As shown in Fig. 12, PED-YOLO performs well in object detection. It achieves 67% bicycle detection accuracy, 62% bus detection accuracy, 87% car detection accuracy, 73% motorcycle detection accuracy, and 81% pedestrian detection accuracy. The confusion matrix also reveals misclassification patterns, providing a basis for further optimization. Despite the robust performance of PED-YOLO, some false detections and missed detections remain. Future improvements could involve optimizing feature extraction, adjusting the loss function, or increasing data diversity.

V. Conclusion

In the field of fog detection, low detection accuracy, complex network model structure, and an excessive number of parameters have long been key challenges. To tackle these issues, a lightweight framework named PED-YOLO is proposed. It integrates modules such as PConv, EfficientNetV2, and ADown to enhance both efficiency and performance. Firstly, the Fused MBConv1, Fused MBConv4, MBConv4, and MBConv6 modules in the EfficientNetV2 network are utilized to replace the C3K2 and Conv modules in the original YOLO11. This modification reduces the weight of the backbone and enhances its feature extraction capability. Secondly, the PConv + ADown network structure is implemented to replace the C3K2 and Conv modules in the original YOLO11 neck network. This effectively decreases the number of parameters and computations during the training process.

In this paper, evaluation methods such as heat maps, normalized confusion matrices, P curves, and R curves are used to compare and analyze the proposed detection framework with other networks, including the original YOLO11. In the heat map detection results, the color of the generated detection area in the PED-YOLO framework is darker. This indicates a significant enhancement in its response intensity to the target. Compared to the original YOLO11 and other network models, the focus region is more accurate. In the normalized confusion matrix, the diagonal values also show improvements across all models. The PED-YOLO framework captures and focuses on the key features of the target in a foggy environment more accurately. As a result, the detection performance is significantly improved.

In addition, compared to the original YOLO11 model, the mAP50 value of the proposed framework on the RTTS dataset reaches 76.7, an increase of 0.92%. The model size is 4.04 MB, a reduction of 23.05%. The number of parameters is 1,886,587, a decrease of 27.18%. The GFLOPs is 4.8, reflecting a reduction of 25%.

In summary, the proposed lightweight PED-YOLO framework significantly improves detection accuracy. At the same time, it effectively simplifies model complexity, reduces model size, and minimizes parameters. This framework provides an efficient and accurate solution for fog detection with excellent performance. It is expected to play an important role in the practical application of fog detection and offer valuable insights for research in related fields.

REFERENCES

[1] D. Cireşan, U. Meier, J. Masci, and J. Schmidhuber, ‘‘A committee of neural networks for traffic sign classification,’’ in Proc. Int. Joint Conf.Neural Netw., San Jose, CA, USA, Jul./Aug. 2011, pp. 1918–1921

[2] D. Ciresan, U. Meier, J. Masci, and J. Schmidhuber, ‘‘Multi-column deep neural network for traffic sign classification,’’ Neural Netw., vol. 32,pp. 333–338, Aug. 2012.

[3] R. Girshick, J. Donahue, T. Darrell, and J. Malik, ‘‘Rich feature hierarchies for accurate object detection and semantic segmentation,’’ in Proc.IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Columbus, OH, USA, Jun. 2014, pp. 580–587.

[4] R. Girshick, ‘‘Fast R-CNN,’’ in Proc. IEEE Int. Conf. Comput. Vis. (ICCV),Dec. 2015, pp. 1440–1448.

[5] S. Ren, K. He, R. Girshick, and J. Sun, ‘‘Faster R-CNN: Towards real-time object detection with region proposal networks,’’ IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, ‘‘You only look once: Unified, real-time object detection,’’ in Proc. IEEE Conf. Comput. Vis.Pattern Recognit. (CVPR), Las Vegas, NV, USA, Jun. 2016, pp. 779–788.

[7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, ‘‘SSD: Single shot multibox detector,’’ in Proc. Eur. Conf.Comput. Vis. (ECCV), vol. 9905. Cham, Switzerland: Springer, Oct. 2016, pp. 21–37.

[8] L. P. Lu, Q. Xiong, D. F. Chu, and B. R. Xu, “MixDehazeNet: Mix Structure Block For Image Dehazing Network,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2023, pp. 1–14.

[9] B. Li, X. Peng, Z. Wang, J. Xu, and D. Feng, “All-in-One Dehazing Network,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 1–9.

[10] F. Zhong, W. Shen, H. Yu, G. Wang, and J. Hu, “Dehazing & Reasoning YOLO: Prior knowledge-guided network for object detection in foggy weather,” Pattern Recognition, vol. 156, pp. 110756, 2024.

[11] F. Akhmedov, R. Nasimov, and A. Abdusalomov, “Dehazing Algorithm Integration with YOLO-v10 for Ship Fire Detection,” Fire, vol. 7, no. 9, p. 332, 2024.

[12] G. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, and G. Ding, “YOLOv10: Real-Time End-to-End Object Detection,” in 38th Conference on Neural Information Processing Systems (NeurIPS 2024), vol. 12372, pp. 1–21.

[13] Z. Chu, “D-YOLO a robust framework for object detection in adverse weather conditions,”2024.

[14] K. Ramesh Babu and P. Venkatesh, “R-YOLO: A Robust Object Detector in Adverse Weather,” Electronics, vol. 12, no. 2, pp. 296–311, 2024.

[15] Donghao Hou, Yujun Zhang, and Jia Ren, "A Lightweight Object Detection Algorithm for Remote Sensing Images," *Engineering Letters*, vol. 33, no. 3, pp704-711, 2025.

[16] S. Zhou, S. Ao, Z. Yang, and H. Liu, “Surface Defect Detection of Steel Plate Based on SKS-YOLO,” Electronics, vol. 12, no. 2, pp. 296–311, 2024.

[17] Jierun Chen, Shiu-hong Kao, Hao He, Weipeng Zhuo, Song Wen, Chul-Ho Lee, S.-H. Gary Chan, “Run, Don’t Walk: Chasing Higher FLOPS for Faster Neural Networks,”arXiv, 2023.

[18] Z. Wu, Y. Zhang, X. Wang, H. Li, Y. Sun, and G. Wang, “Algorithm for detecting surface defects in wind turbines based on a lightweight YOLO model,” in Proc. Eur. Conf. Comput. Vis. (ECCV), vol. 1437. Berlin, Germany: Springer, Dec. 2024, pp. 1-13.

[19] J. Zhang, X. Wei, L. Zhang, L. Yu, Y. Chen, and M. Tu, “YOLO v7-ECA-PConv-NWD Detects Defective Insulators on Transmission Lines,” in Proc. Eur. C.

[20] Mingxing Tan, Quoc Le.EfficientNetV2: Smaller Models and Faster Training. Proceedings of the 38th International Conference on Machine Learning, PMLR 139:10096-10106, 2021.

[21] T. Yin, W. Chen, B. Liu, C. Li, and L. Du, “Light “You Only Look Once”: An Improved Lightweight Vehicle-Detection Model for Intelligent Vehicles under Dark Conditions,” in Proc. Eur. Conf. Comput. Vis. (ECCV), vol. 12, no. 124. Cham, Switzerland: Springer, Dec. 2023, pp. 1-19.

[22] J. Hu, L. Shen, and G. Sun, “Squeeze-and-Excitation Networks,” in Proc. Eur. Conf. Comput. Vis. (ECCV), vol. 10, no. 4. Cham, Switzerland: Springer, Sep. 2017, pp. 7132-7141.

[23] J. Li, Y. Chen, M. Niu, W. Cai, and X. Qiu, “ADS-YOLO: A Multi-Scale Feature Extraction Remote Sensing Image Object Detection Algorithm Based on Dilated Residuals,” in Proc. Eur. Conf. Comput. Vis. (ECCV), vol. 13, no. 2. Cham, Switzerland: Springer, Feb. 2025, pp. 26225-26234.

[24] M. Zhou, X. Wan, Y. Yang, J. Zhang, S. Li, S. Zhou, and X. Jiang, “EBR-YOLO: A Lightweight Detection Method for Non-Motorized Vehicles Based on Drone Aerial Images,” in Proc. Eur. Conf. Comput. Vis. (ECCV), vol. 25, no. 1. Cham, Switzerland: Springer, Jan. 2025, pp. 196-211.

[25] V. A. Sindagi and P. Oza, “Prior-Based Domain Adaptive Object Detection for Hazy and Rainy Conditions,” in Computer Vision – ECCV 2020. Cham, Switzerland: Springer, 2020, vol. 12372, pp. 723–739.

[26] Y. Xie, Y. Xie, L. Chen, C. Li, and Y. Qu, "Object detection in real foggy scenes," Journal of Computer-Aided Design & Computer Graphics, vol. 33, no. 5, pp. 734-745, Beijing: China Computer Federation, 2021.

[27] Tong Zhou, Xiaoxia Zhang, and Huilong Chen, "A Dangerous Driving Behavior Detection Method Based on Improved YOLOv8s," *Engineering Letters*, vol. 33, no. 3, pp721-731, 2025.

[28] S. Zhou, K. Cai, Y. Feng, X. Tang, H. Pang, J. He, and X. Shi, “An Accurate Detection Model of Takifugu rubripes Using an Improved YOLO-V7 Network,” in Journal of Marine Science and Engineering, vol. 11, no. 5. Basel, Switzerland: MDPI, May 15, 2023, p. 1051