Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.058474

ARTICLE

Enhanced Multi-Scale Object Detection Algorithm for Foggy Traffic Scenarios

Honglin Wang[1], Zitong Shi[2,*] and Cheng Zhu[3]

[1]School of Artificial Intelligence, Nanjing University of Information Science and Technology, Nanjing, 210044, China

[2]School of Computer Science, Nanjing University of Information Science and Technology, Nanjing, 210044, China

[3]School of Electrical & Computer Engineering, University of Illinois at Urbana Champaign, Urbana, IL 61801, USA

*Corresponding Author: Zitong Shi. Email: 202312490280@nuist.edu.cn

Received: 13 September 2024 Accepted: 20 November 2024 Published: 17 February 2025

ABSTRACT

In foggy traffic scenarios, existing object detection algorithms face challenges such as low detection accuracy, poor robustness, occlusion, missed detections, and false detections. To address this issue, a multi-scale object detection algorithm based on an improved YOLOv8 has been proposed. Firstly, a lightweight attention mechanism, Triplet Attention, is introduced to enhance the algorithm's ability to extract multi-dimensional and multi-scale features, thereby improving the receptive capability of the feature maps. Secondly, the Diverse Branch Block (DBB) is integrated into the CSP Bottleneck with two Convolutions (C2F) module to strengthen the fusion of semantic information across different layers. Thirdly, a new decoupled detection head is proposed by redesigning the original network head based on the Diverse Branch Block module to improve detection accuracy and reduce missed and false detections. Finally, the Minimum Point Distance based Intersection-over-Union (MPDIoU) is used to replace the original YOLOv8 Complete Intersection-over-Union (CIoU) to accelerate the network's training convergence. Comparative experiments and dehazing pre-processing tests were conducted on the RTTS and VOC-Fog datasets. Compared to the baseline YOLOv8 model, the improved algorithm achieved mean Average Precision (mAP) improvements of 4.6% and 3.8%, respectively. After defogging pre-processing, the mAP increased by 5.3% and 4.4%, respectively. The experimental results demonstrate that the improved algorithm exhibits high practicality and effectiveness in foggy traffic scenarios.

KEYWORDS

Deep learning; object detection; foggy scenes; traffic detection; YOLOv8

1 Introduction

In the foggy scenes, due to low light intensity and the presence of water droplets or suspended particles in the atmosphere, the visibility of objects within the field of view can be significantly compromised. This results in blurred or missing targets in captured images, reduced contrast, and unclear feature information. Such conditions pose significant challenges to the performance of object detection systems and present considerable difficulties for computer vision systems. Nevertheless, advancements in object detection technology are particularly critical under these challenging conditions. In fields such as urban traffic monitoring, autonomous vehicles, and security surveillance, the

ability of detection systems to identify objects in foggy conditions directly impacts the reliability and safety of daily human activities. In traffic monitoring systems, adverse weather conditions can lead to frequent traffic accidents, so, making the timely and accurate detection of vehicles and pedestrians on the road is crucial. Therefore, studying how to enhance the precision and robustness of object detection in foggy scenarios is of great practical significance.

In the past few years, the problem of object detection under foggy scenes has been a prominent research topic in the field of computer vision. Traditional approaches mainly rely on conventional computer vision techniques, although these methods demonstrate some effectiveness in processing images under foggy conditions, their performance is not particularly robust when applied to real-world foggy traffic scenarios, where the conditions are complex and variable. The rapid advancements in deep learning and neural network technologies provide novel insights and methodologies for addressing challenges in this field. Unlike traditional methods, deep learning technologies can directly extract data features from raw datasets, demonstrating superior robustness and generalization capabilities. However, deep learning technology is heavily reliant on datasets, and most of the datasets used by current object detection models consist of high-resolution images with clear quality. In tasks requiring detection under complex environmental conditions, such as adverse weather, the low imaging quality of the images makes it challenging to collect relevant, real, and reliable datasets. Consequently, directly applying these models to such scenarios often fails to achieve satisfactory detection results, because in the real-world environments, various weather conditions and target information to be detected will affect each other during imaging, resulting in a decrease in imaging quality. This interaction causes the targets to be difficult to distinguish from the background, leading to the decreased detection performance.

Currently, methods to address the aforementioned issues mainly fall into two categories. The first category comprises unrelated dehazing and detection models, which dehazing is performed first, followed by object detection. This approach uses image dehazing or enhancement methods to preprocess the input images, removing weather effects and noise, and then feeds the preprocessed images into the object detection model for training and inference. For example, Li et al. [1] proposed an unrelated dehazing object detection method combining the dehazing network AOD-Net with Faster R-CNN [2]. However, such methods often require the use of complex image restoration networks, which can impact the accuracy of the object detection model. The second category involves the joint optimization of dehazing algorithms and object detection algorithms, performing dehazing and detection simultaneously. For instance, Huang et al. [3] employed two subnetworks for joint learning to accomplish both object detection and image restoration tasks. Since these networks share feature layers, it becomes challenging to balance the weights of the detection and dehazing tasks by adjusting parameters during training. Although this approach improves the accuracy of object detection in foggy scenarios, the dehazing effect may be suboptimal, often resulting in artifacts in the dehazed images.

In foggy traffic scenarios, issues such as poor visibility, reduced light, and blurred object edges can lead to a decline in algorithm performance. Traditional object detection algorithms often underutilize image features, easily overlooking edge features. Additionally, the low pixel proportion of small targets in foggy scenes can result in missed detections. In practical detection, occlusion often leads to false detections and missed detections, making these algorithms unsuitable for real-world object detection tasks. Furthermore, foggy traffic scenarios detections demand high real-time performance for the models, the complex detection environment, reduced target saliency, and difficulty in extracting feature information affect detection accuracy and degrade system robustness. To address these challenges and accomplish this task, at least two prerequisites must be met: (1) the model should be lightweight (2) the model's accuracy should be maximized. Therefore, we use YOLOv8, which is known for its high

accuracy and speed, as the baseline model. YOLOv8 performs excellently across different scenarios and datasets, exhibiting strong generalization capabilities. In this paper, our work is improving the YOLOv8s network model to make it more suitable for detection in foggy scenarios. The specific improvement process is detailed in the Section 4. We validate our improved model algorithm on the RTTS [4] real foggy dataset and the VOC-Fog synthetic dataset, both of which show significant improvements in average precision.

The main contributions of this paper can be summarized in the following five aspects:

- In the network's backbone and neck sections, we introduce the multidimensional, multiscale lightweight attention mechanism Triplet Attention to further extract multilevel, multiscale, and information-rich image features. This approach enhances the semantic representation, aiming to improve scene recognition accuracy.
- We utilized the residual diverse branch block to redesign the original baseline model's C2F module and developed a new multi-scale feature enhancement module. This module aims to strengthen the multi-scale fusion of feature information across different levels. By fully leveraging the attention mechanism and multi-scale information of the backbone feature extraction network, it highlights significant channel-level information features, which has a positive impact on enhancing feature representation capabilities.
- A new detection head has been proposed, DetectEfficientHead, which is redesigned based on the residual diverse branch block. By stacking and integrating two consecutive DBB modules, it replaces the previous two standard convolutional layers to enable parameter sharing. This approach increases the overall depth of the detection head, capturing more contextual information to improve detection performance and reduce false detections and missed detections.
- We employ the MPDIoU loss function to accelerate the convergence speed of loss reduction during training and enhance the network's inference capability, which significantly improves the precision and speed of bounding box localization while maintaining computational efficiency.
- Using the Light-DehazeNet dehazing algorithm for preprocessing the RTTS and VOC-Fog datasets to better demonstrate the detection performance of our proposed method. The improved model achieves excellent scene recognition accuracy on both challenging datasets, outperforming several mainstream foggy weather detection algorithms and state-of-the-art detection algorithms in terms of performance metrics.

The rest of this paper is organized as follows: Section 2 introduces related work on object detection and image dehazing. Section 3 introduces the research method in detail. Section 4 describes the experimental settings and experiment results analysis. In Section 5, the content of this paper is summarized and conclusions are presented.

2 Related Works

2.1 Object Detection

Current mainstream deep learning-based object detection methods can be broadly categorized into two-stage detection algorithms, such as the R-CNN [5] series, including Fast R-CNN [6] and Mask R-CNN [7]. These algorithms typically require more computational resources but often excel in precision, particularly in complex scenarios and small object detection. For instance, Hu et al. [8] extended the Faster R-CNN by incorporating three domain classifiers, which help the network extract domain-invariant features between the source domain (normal weather) and the target domain (rainy or foggy weather). These classifiers address domain discrepancies from three angles: local image level,

global image level, and instance level. Similarly, Chen et al. [9] proposed a domain adaptation method that performs features alignment and domain adaptation between the source and target domains, thereby enhancing detection accuracy in the target domain. However, due to the high computational resource requirements and associated costs, these methods are less suitable for real-time scenarios with strict time constraints. Ge et al. [10] proposed a neural attention learning method that does not require the addition of extra modules. This approach generates attention response maps by backtracking the predictions of convolutional neural networks (CNNs), to diagnose the network's state. These response maps are utilized to optimize the loss function, enabling the network to focus more on foreground objects, thereby improving object detection accuracy. This method dynamically adjusts the model's focus, enhancing detection capabilities in complex scenes. However, the introduction of neural attention mechanisms may also increase computational resource consumption, particularly when dealing with complex datasets.

Despite the accuracy advantages of the aforementioned methods, single-stage detection architectures like YOLO [11] and SSD [12] offer faster inference speeds due to the absence of an additional region proposal stage, making them more suitable for rapid deployment in practical applications. As end-to-end detection methods, they directly predict both object categories and locations, featuring strong real-time performance and a simple efficient structure.

For example, the literature [13] demonstrates using the SSD object detection algorithm as a baseline model combined with the AOD-Net dehazing network for detection in foggy scenarios. This method enhances object detection performance in foggy conditions through image preprocessing; However, it demonstrates limitations in multi-scale feature extraction and fusion within the detection network, failing to fully exploit multi-dimensional information. In contrast, we incorporate the Triplet Attention mechanism into the backbone and neck sections to enhance multi-dimensional and multi-scale feature extraction capabilities. This not only provides richer image features, but also significantly improves scene recognition accuracy with minimal computational complexity. Particularly in complex foggy scenarios, our method captures more detailed information, thereby improving overall detection performance.

Another literature [14] incorporated an image enhancement algorithm from generative adversarial networks into the preprocessing module of YOLOv4 to better preserve the high-quality textures and feature information of the dehazed images. Although this method improves the preservation of image details, the introduction of GANs may lead to instability during the training process and has limited capacity for capturing contextual information. In our method, we address these issues by designing a novel decoupled head, DetectEfficientHead, which leverages the DBB module for efficient context information capture, replacing traditional convolutional layers. This not only increases the structural depth of the detection head but also ensures comprehensive sharing of contextual information. Compared to the YOLOv4 approach combined with GANs, our innovative use of DBB enhances the detector's robustness in feature extraction and context awareness, thereby reducing false positives and missed detections.

In another example, the IA-YOLO [15] model includes a differentiable image processing module, utilizing image adaptive techniques to eliminate adverse weather effects and recover the underlying content. This method adjusts image quality based on the specific weather conditions of the input image, enhancing the image adaptively in adverse weather conditions like fog for better object detection. However, this approach introduces unwanted noise into the object detector. We employed Light-DehazeNet for dehazing preprocessing, which not only ensures effective dehazing but also reduces noise introduction due to the algorithm's strong generalization ability. When combined with

our improved detection algorithm, the dehazed images fully leverage the feature extraction advantages of the Triplet Attention and DBB structures.

Hnewa et al. [16] proposed a cross-domain object detection method that uses multi-scale features and domain adaptation techniques to improve detection performance in adverse weather conditions. Although this method improves detection performance in cross-domain scenarios, it may still be insufficient in the fusion and extraction of multi-scale information. Our method addresses this by utilizing an improved C2F module with a residual DBB structure, proposing a new multi-scale feature enhancement module that significantly improves the fusion of features at different scales. Compared to the cross-domain detection approach by Hnewa et al., our module design more effectively exploits multi-scale feature information. Coupled with the Triplet Attention mechanism, it ensures the capture of prominent channel-level features at various layers, resulting in a substantial improvement in detection accuracy and robustness.

2.2 Image Dehazing

Image dehazing technology is a technique designed to address image blurring and reduced contrast caused by atmospheric interference factors such as haze and fog. It is a crucial task in computer vision, aiming to restore image clarity and details through various algorithms and methods, thereby enhancing the readability and usability of images. Currently, commonly used methods in the fields of dehazing and object detection include the following:

1. Image Enhancement-Based Dehazing Methods:

Histogram Equalization [17]: This method involves statistically analyzing the original histogram of the input image to obtain appropriate values, followed by stretching the image to improve contrast. The goal is to increase the grayscale difference between pixels in the image, thereby enhancing its clarity.

Retinex Algorithm [18]: Based on Retinex theory, this method extracts the reflection component of the image and performs enhancement processing to achieve image improvement. making the image clearer.

Overall, the dehazing method based on image enhancement is relatively simple in terms of technical difficulty, but after processing, the image may suffer from serious distortion problems, resulting in poor dehazing effect.

2. Physics-Based Dehazing Methods:

A classical method in this category is the Atmospheric Scattering Model [19], initially proposed by McCartney and later theoretically derived Narasimhan et al. [20].This theory posits that reduced visibility in foggy conditions is due to the scattering and absorption of sunlight by suspended particles in the air, which leads to transmission attenuation in the captured image. The atmospheric scattering model is expressed by the following formula:

$$I(x) = J(x)t(x) + \alpha(1 - t(x)) \tag{1}$$

$$t(x) = e^{-\beta d(x)} \tag{2}$$

In Eq. (1), $I(x)$ represents the blurred image captured by the camera, while $J(x)$ denotes the scene radiance that can be considered as the unblurred image. In Eq. (2), β represents the atmospheric scattering coefficient, and $d(x)$ denotes the scene depth.

Dark Channel Prior Algorithm [21]: This algorithm is based on the atmospheric scattering model theory, this research discovered that in most haze-free images, there are always regions where the pixel values in at least one color channel are very low, and the grayscale values tend to approach zero. The algorithm learns and analyzes certain feature information from a large number of haze-free images to identify a prior relationship between the clean image and the atmospheric scattering model, ultimately achieving dehazing. While this algorithm processes images quickly, its effectiveness can be limited in specific scenarios.

3. Deep Learning-Based Dehazing Methods Using Convolutional Neural Networks:

This approach is relatively straightforward and popular, involving the use of convolutional neural networks (CNNs) to establish an end-to-end model that directly restores haze-free images from hazy ones. Cai et al. [22] proposed the DehazeNet dehazing network model, which utilizes CNNs to learn the direct mapping between hazy images and their transmission maps. But this dehazing algorithm has high computational complexity, particularly for large images, and may not perform well with low-contrast images. DehazeNet is also sensitive to noise, potentially introducing artifacts or pseudo-haze effects during processing. Li et al. [1] introduced the end-to-end integrated dehazing network AOD-Net, which reconstructs and deforms the atmospheric scattering model to directly generate haze-free images from hazy inputs, without relying on intermediate parameter estimation steps, this method simplifies the dehazing process and demonstrates significant effectiveness.

The dehazing algorithm employed in this paper is Light-DehazeNet [23]. The selection of this algorithm is based on the following three reasons:

(a) Accuracy: By learning from a large number of training samples, Light-DehazeNet can effectively remove haze while preserving the original details and colors of the image, thus enhancing visual quality.

(b) Strong Generalization: Light-DehazeNet is capable of handling various types of hazy images, exhibiting excellent generalization performance.

(c) High Efficiency: Traditional dehazing algorithms often suffer from high computational complexity and unstable performance. Light-DehazeNet features a compact network structure that reduces computational resource requirements while maintaining high dehazing performance. Its lightweight design provides a significant advantage in terms of processing speed, making it well-suited for real-time applications.

3 Methods

We first introduce a lightweight attention mechanism Triplet Attention [24], between the backbone feature extraction network and the neck part of the network. This is to ensure that while enhancing the model performance, the model remains lightweight and its complexity is reduced. The tri-branch structure of this attention mechanism better captures cross-dimensional interactive features, thereby enhancing the network's feature extraction capabilities and facilitating the delivery of detailed image features to the neck part for feature fusion. To further improve the model's accuracy, especially in achieving higher precision recognition in foggy scenarios, we replace the Conv layer in the Bottleneck of C2F with a Diverse Branch Block [25] to form a new module, CSP Bottleneck with two Convolutions-Diverse Branch Block (C2F-DBB). Additionally, based on the DBB module, we redesign a lightweight detection head, named DetectEfficientHead, to perform better classification and prediction tasks. This reduces the rates of missed and false detections. Furthermore, we replace

the CIOU loss function [26] with the MPDIoU loss function [27] to accelerate the loss convergence speed and improve the model inference speed. The improved network structure shown in Fig. 1.

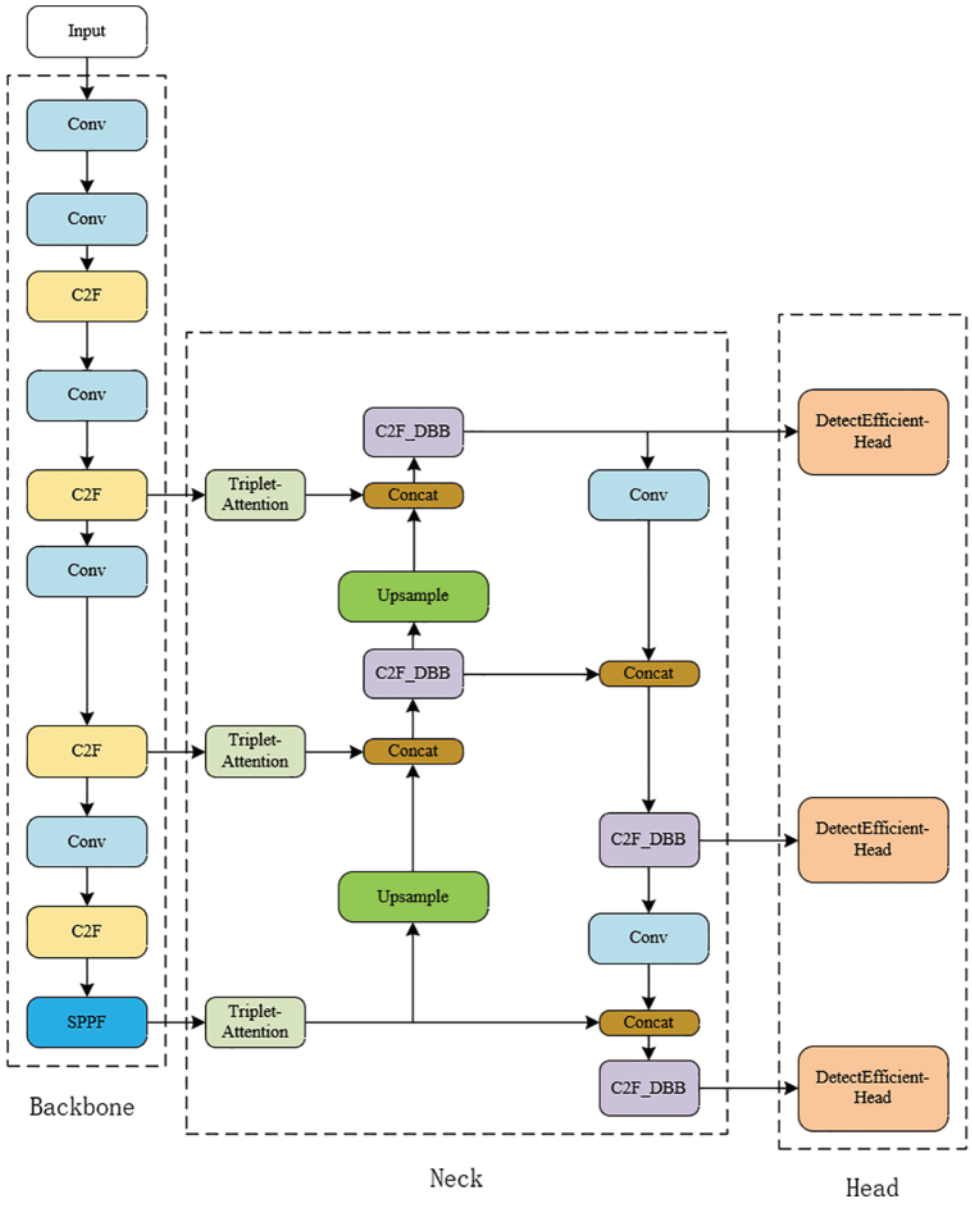


Figure 1: The network structure of our method

3.1 Triplet Attention Mechanism

The traditional method for computing channel attention mainly aims to calculate the weights of channels. The input tensor is decomposed into a single pixel through global average pooling in the spatial dimension, which results in the loss of substantial information during training. Consequently, the interdependence between the channel and spatial dimensions is also lost when calculating attention on a single-pixel channel. Although the Convolutional Block Attention Module (CBAM) proposed by

Woo et al. [28] alleviates the issue of spatial interdependence by incorporating both spatial and channel attention, the channel attention and spatial attention in CBAM are still computed separately and independently. Chen et al. [29] aims to directly perform referring expression object segmentation from compressed videos. Through a dual-path, dual-attention module, the network can process different video data modalities in parallel to extract more robust feature representations. These methods dynamically adjust the model's focus, enhancing its adaptability in various applications. However, the introduced attention mechanisms may also increase computational resource consumption, especially when handling complex datasets. For the choice of attention mechanisms in our study's application scenario, the attention module should be lightweight and efficient.

To address this, the Triplet Attention mechanism introduces the concept of cross-dimensional interactions based on the method of establishing spatial attention. The Triplet Attention mechanism consists of three branches, each responsible for aggregating the interaction features between specific dimensions and the channel dimension of the input tensor. By employing rotation and residual connection operations, the interdependencies between dimensions are established. This mechanism effectively captures the interactive features among different dimensions of the input, thereby enhancing the understanding of image content. The Triplet Attention structure is shown in Fig. 2.

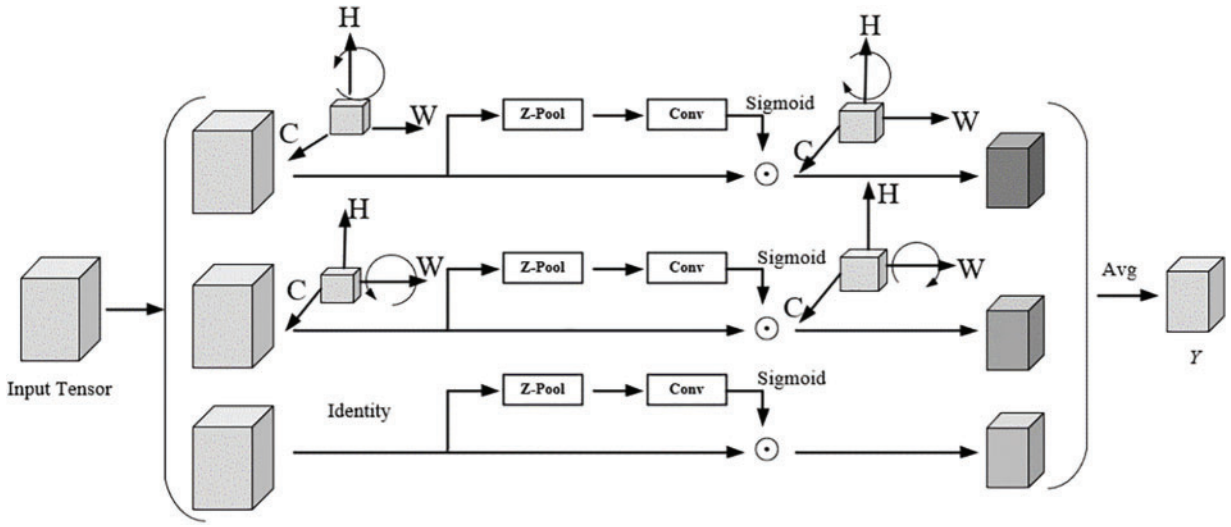


Figure 2: The Triplet Attention structure diagram

Specifically, for an input tensor $\lambda \in R^{C\times H\times W}$, the main process first involves passing the input tensor to each branch for operation. In the first two branches, the first branch is responsible for calculating the attention weights across the channel and spatial dimensions, handling the interactive features between the spatial and channel dimensions of the input. In the middle branch, the attention weights for both the channel and spatial dimensions are computed, while also processing the interaction features between these dimensions. The main process involves rotating the width or height of the input tensor λ 90 degrees counterclockwise, then passing the rotated tensor $\widehat{\chi_1}$ or $\widehat{\chi_2}$ through a Z-Pool layer to reduce the channel dimension to 2D. The tensor at this stage is denoted as $\widehat{\chi_1^*}$ or $\widehat{\chi_2^*}$. The average pooling features and max pooling features of this dimension are then concatenated. This allows the layer to retain a rich representation of the actual tensor while reducing its depth, thus further lowering

the computational load and accelerating training. The expression for Z-Pool is given in Eq. (3):

$$Z - \text{Pool}(\chi) = [\text{MaxPool}_{0d}(\chi), \text{AvgPool}_{0d}(\chi)] \tag{3}$$

Then, the tensor is passed through a $K \times K$ convolutional layer and a batch normalization layer. The output is then activated by a sigmoid activation function, generating attention weights with dimensions $1 \times H \times W$. Finally, the tensor is rotated 90 degrees clockwise along the width or height, corresponding to the first step, ensuring that the output tensor has the same dimensions as the input tensor.

The final branch at the bottom is designed to capture spatial dependencies, specifically to obtain the interactive features of the spatial dimensions. Similar to CBAM, it constructs spatial attention. The specific process is identical to the previous two branches, except for the rotation operation. Finally, the output tensors of the three branches are aggregated and averaged to obtain the final cross-dimensional interaction feature y. The calculation expression for y is shown in Eq. (4):

$$y = \frac{1}{3}\left(\overline{\widehat{\chi}_1 \sigma\left(\psi_1\left(\widehat{\chi}_1^*\right)\right)} + \overline{\widehat{\chi}_2 \sigma\left(\psi_2\left(\widehat{\chi}_2^*\right)\right)} + \chi\sigma\left(\psi_3\left(\widehat{\chi}_3\right)\right)\right) \tag{4}$$

In Eq. (4), σ represents the sigmoid activation function; $\Psi 1$, $\Psi 2$ and $\Psi 3$ denote the standard 2D convolution layers in the three branches of triplet attention, defined by kernel size k. $\widehat{\chi}_1$ and $\widehat{\chi}_2$ represent the tensors after rotation in the first two branches, while $\widehat{\chi}_1^*$, $\widehat{\chi}_2^*$ and $\widehat{\chi}_3$ represent the tensors after passing through the Z-Pool layer, λ denotes the original input tensor.

In foggy traffic scenarios, high accuracy and real-time performance are crucial for object detection. Therefore, the model's parameter complexity and overhead need to be important considerations. From Table 1 (where C represents the number of input channels to the layer, r denotes the reduction ratio used at the bottleneck when computing channel attention between neural network layers, and k represents the kernel size used for 2D convolution), it can be concluded that compared to other standard attention mechanisms, the computational overhead of triplet attention is significantly lower. This is because it focuses only on the interactions among three elements rather than computing the entire input, thereby reducing the computational load and lowering the time complexity. For this reason, it can be easily extended to larger input sizes, allowing it to operate on larger inputs while maintaining a low computational complexity. All these factors validate the efficiency of triplet attention.

Table 1: Comparison of different attention mechanisms

Attention mechanism	Parameters	Overhead
Squeeze-and-Excitation mechanism (SE) [30]	$2C^2/r$	2.514 M
CBAM [28]	$2C^2/r+2k^2$	2.532 M
Bottleneck Attention Module (BAM) [31]	$C/r\ (3C+2k^2C/r+1)$	0.358 M
Global Correlation Network (GC) [32]	$2C^2/r+C$	2.548 M
Triplet Attention [24]	$6k^2$	0.0048 M

3.2 Residual Diverse Branch Block

Residual Diverse Branch Block is a structural re-parameterization technique that can enhance the performance of convolutional neural networks. This multi-branch structure, similar to Inception

module, based on this, we introduced residual links with the primary aim of mitigating the gradient vanishing problem during training, improving the training speed and accuracy of the network, enhancing the network's expressive power, and supporting the construction of deep networks. An example of the module design is shown in Fig. 3:

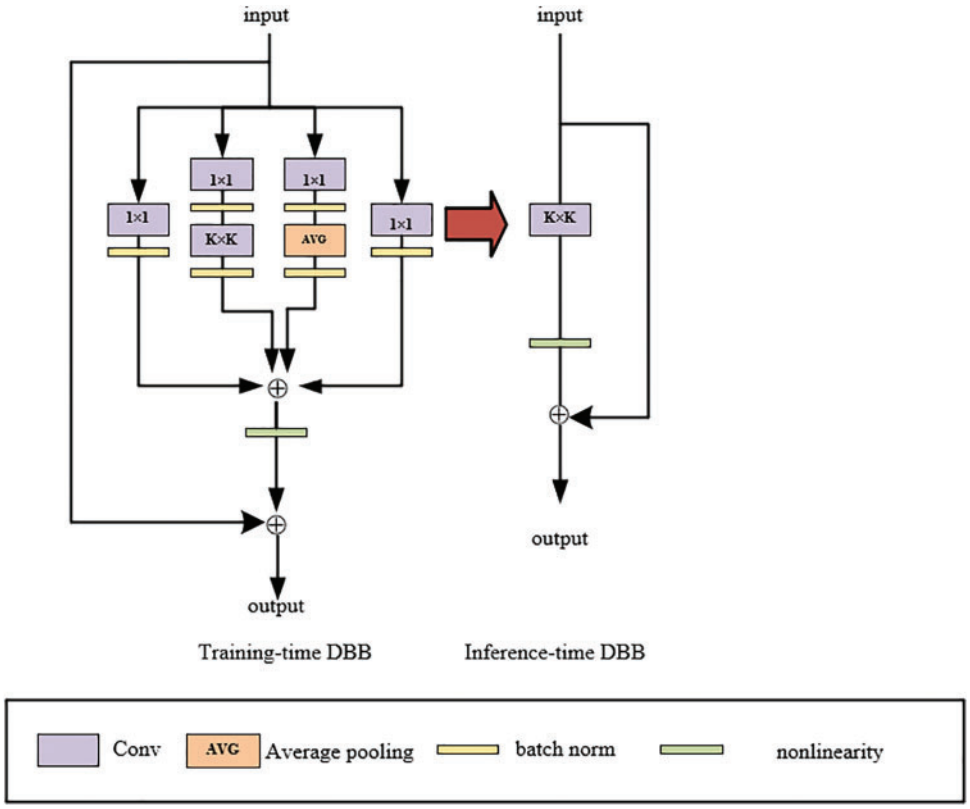


Figure 3: Residual Diverse Branch Block structure design diagram

The Residual Diverse Branch Block can enhance the performance of convolutional networks while not increasing any inference time. During the training phase, it employs a complex branch structure, combining branches of different scales and complexities (with different branches utilizing convolution sequences, multi-scale convolutions, or average pooling) to enrich the diversity of the feature space, thereby enhancing the expressive capability of a single convolution (merged into a single convolution during inference). Thus, it can be seamlessly integrated into any existing architecture as a substitute. Once training is complete, a DBB can be equivalently converted into a single convolution during the inference phase for easier deployment while maintaining efficient inference. In this way, the model can be trained to achieve higher performance levels. In the DBB, a convolution operation can be expressed as shown in Eq. (5):

$$O = I * F + REP(b) \tag{5}$$

the symbol $*$ denotes the convolution operator, $I \in R^{C\times H\times W}$ represents the input tensor, $O \in R^{C'\times H'\times W'}$ represents the output tensor, $F \in R^{D\times C\times K\times K}$ and is the convolution kernel. For the convenience of subsequent merging, the bias parameter is denoted as $REP(b) \in R^{D\times H'\times W'}$.

The value at the j-th output channel position (h, w) can be given by Eq. (6), where $X(c, h, w)_{u,v} \in R^{K\times K}$ denotes a sliding window on the C-th channel of the input frame I, corresponding to the coordinates (h, w) of the output frame.

$$O_{j,h,w} = \sum_{c=1}^{C} \sum_{u=1}^{K} \sum_{v=1}^{K} F_{j,c,u,v} X(c, h, w)_{u,v} + b_j \tag{6}$$

From Eq. (6), it can be deduced that the convolution operation possesses homogeneity and additivity, as specifically expressed in Eqs. (7) and (8). Additionally, the condition for additivity to hold is that the two convolution operations have the same configuration (number of channels, kernel size, stride, and padding, etc.).

$$I * (pF) = p(I * F) \tag{7}$$

$$I * F^{(1)} + I * F^{(2)} = I * \left(F^{(1)} + F^{(2)}\right) \tag{8}$$

As shown in Fig. 4, the DBB used during training can be converted into a conventional convolution layer used for inference through six different transformation methods. Each transformation method corresponds to a specific operation. These six operations include Conv-BN merging, parallel merging, serial merging, parallel concatenation, average pooling transformation, and multi-scale convolution merging. During the model inference phase, the model structure is simplified to convolution layers. This design allows the model to leverage the diversity of the diverse branch block to enhance feature extraction and learning effects during training, while in real-world applications, i.e., during inference, it achieves efficient operation by reducing computational complexity. This module maintains excellent performance while ensuring efficient computational speed and resource utilization. Therefore, it meets the requirements for real-time object detection in foggy scenarios.

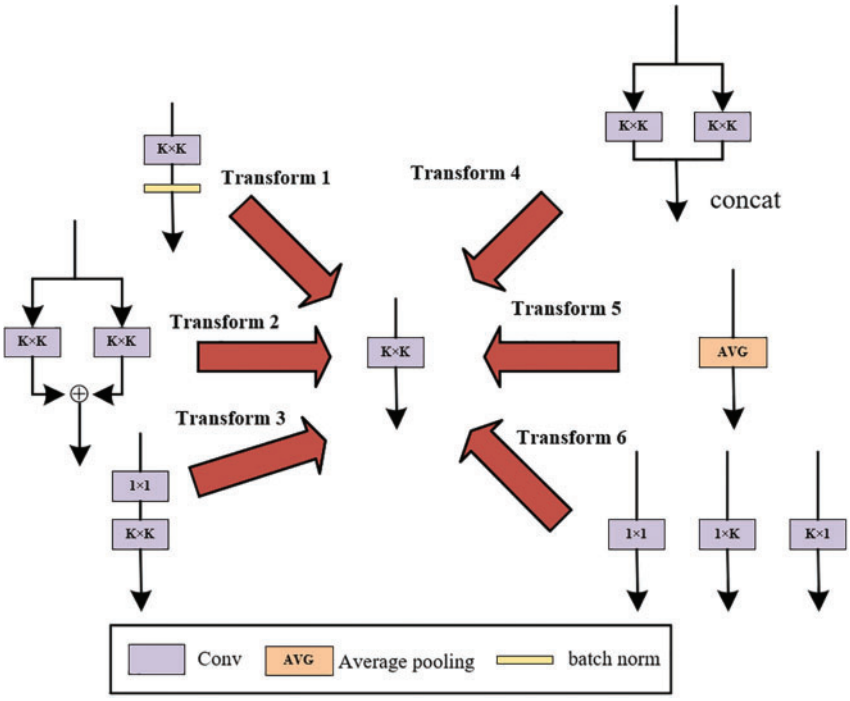


Figure 4: The six Transformations included in DBB

3.2.1 C2F-DBB

In the Bottleneck of the baseline model C2F module, only two traditional large 3×3 convolutional layers are used for feature extraction. In our work, these two conventional convolutional layers in the Bottleneck are replaced with the DBB module to enhance the representation capability of a single convolution. The DBB merges branches of varying complexities and scales, enhancing the feature space. Research has demonstrated that combining branches with different capacities is more effective than combining two branches with identical high capacities (e.g., replacing two 3×3 convolutions with 1×1 conv and 3×3 conv), which inspired the design of C2F-DBB. The specific structure of C2F-DBB is illustrated in Fig. 5. The improvements to the C2F module offer the following two main advantages: 1. Increasing the structural complexity during training to enhance model performance; 2. During inference, it can be equivalently transformed into a simpler structure (i.e., maintaining performance after training).

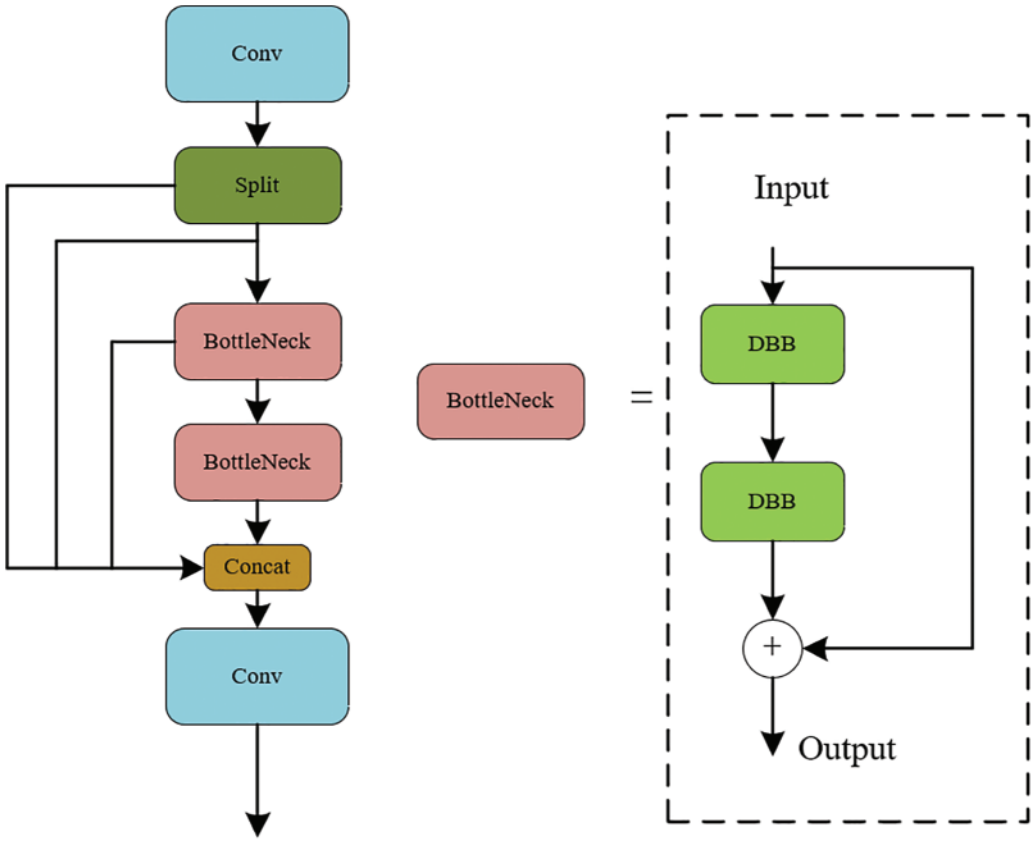


Figure 5: C2F-DBB module structure diagram

3.2.2 DetectEfficientHead

Based on the advantages of the DBB module, our work introduces the DBB module into the head section of the model, redesigning a new detection classification decoupling head called DetectEfficientHead. The original baseline model's detection head has two branches, each containing two convolutional layers and one conv2d layer. The first convolutional layer of both branches is replaced with DBB to better facilitate parameter sharing. Similarly, the second convolutional layer

of both branches is also replaced with DBB, allowing the detection head to better capture contextual information of objects, enhancing the feature representation capability of the convolutional layers, and enabling the network to extract more detailed information about the targets. Finally, the output is fed into a conv layer for prediction tasks and loss computation. Although the use of a multi-branch structure increases the number of parameters and computational time, the computational complexity is reduced during the inference phase after training due to network fusion. The structure of DetectEfficientHead is illustrated in Fig. 6.

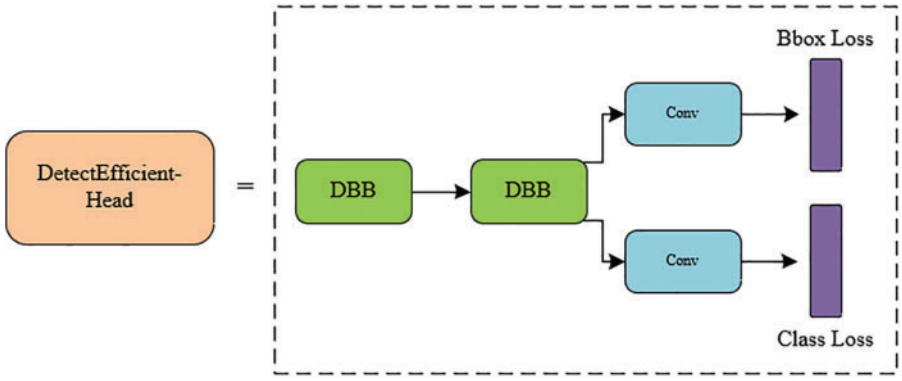


Figure 6: DetectEfficientHead module structure diagram

Both C2F-DBB and DetectEfficientHead are improvements based on the Residual Diverse Branch Block module. The design goal is to maintain a lightweight model while enhancing the detection performance of low-resolution objects in foggy traffic scenarios and generating more comprehensive feature representations. The Residual Diverse Branch Block module enhances the model's generalization, increases its robustness. This adaptability facilitates easier deployment of the model on edge devices, thereby amplifying its real-time performance.

3.3 MPDIoU

In foggy traffic scenarios, enhancing detection accuracy is critically important. YOLOv8 employs CIoU as its loss function, which integrates three key geometric factors: overlap area, centroid distance, and aspect ratio, thus improving the localization of predicted frames. However, traditional CIoU involves several techniques when dealing with the prediction boxes and ground truth boxes, increasing computational complexity during training. It does not account for the IoU of low-resolution small objects in the image, which can lead to sample imbalance issues. Additionally, when the predicted box and the ground truth box have the same aspect ratio but differ significantly in width and height, CIoU may hinder the model's ability to effectively optimize similarity. Therefore, there is considerable room for improvement in CIoU.

To address this, we introduce a new loss function MPDIoU (Minimum Pairwise Distance Intersection over Union), which simplifies the similarity comparison between two bounding boxes by utilizing more geometric constraint information. It helps reduce adjustments to the bounding box positions and sizes during training and ensures that, in cases such as non-overlapping centroids, the predicted box is closer to the ground truth box, thereby improving detection accuracy. This approach addresses the limitations of the CIoU function. It also stabilizes model convergence and enhances detection accuracy for multi-scale targets, The schematic calculation of MPDIoU is illustrated in Fig. 7.

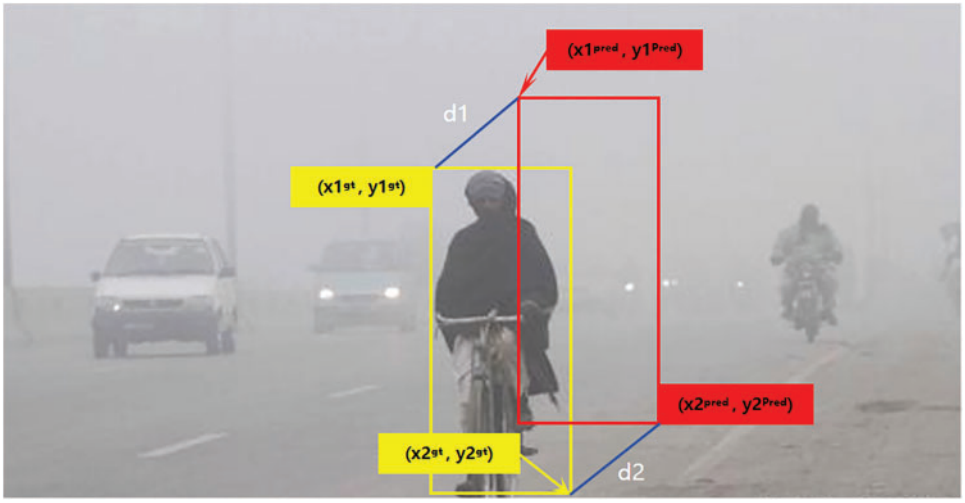


Figure 7: Schematic calculation of MPDIoU

The calculation formula for MPDIoU is as follows:

$$IoU(pbb, gbb) = \frac{\text{Area}(pbb \cap gbb)}{\text{Area}(pbb \cup gbb)} \tag{9}$$

$$d_1^2 = (x_1^{pred} - x_1^{gt})^2 + (y_1^{pred} - y_1^{gt})^2 \tag{10}$$

$$d_2^2 = (x_2^{pred} - x_2^{gt})^2 - (y_2^{pred} - y_2^{gt})^2 \tag{11}$$

$$MPDIoU = IoU - \frac{d_1^2 + d_2^2}{h^2 + w^2} \tag{12}$$

$$L_{MPDIoU} = 1 - MPDIoU \tag{13}$$

Eq. (9) represents Intersection over Union. Since its introduction, the goal of bounding box regression has been to refine the detection window output by the detector to approach the true detection window. IoU has become a mainstream standard for evaluating the loss of predicted boxes in the detection field. *pbb* represents the predicted bounding box, and *gbb* represents the ground truth bounding box. In Eqs. (10) and (11), (x_1^{gt}, y_1^{gt}) and (x_2^{gt}, y_2^{gt}) denote the coordinates of the top-left and bottom-right points of the ground truth, while (x_1^{pred}, y_1^{pred}), (x_2^{pred}, y_2^{pred}) represent the coordinates of the top-left and bottom-right points of the predicted box. d_1^2, d_2^2 respectively represent the euclidean distances between Points 1 and 2 of the predicted box and the ground truth. w, h denote the width and height of the input image.

As can be seen from the above equations, IoU is very sensitive to cases where the bounding boxes partially overlap but have significantly different areas. This can lead to a large number of errors, and calculating it requires finding the intersection and union areas of the two bounding boxes, involving complex geometric computations with high computational complexity, especially when the bounding boxes partially overlap. MPDIoU more accurately reflects the relative positional relationship between two bounding boxes by calculating the distance between their center points. MPDIoU considers the shape and position of the bounding boxes during the calculation process, reducing such errors to some extent and improving the overall robustness of the model. In cases where the bounding boxes are

very close in position but do not completely overlap in shape, this method provides higher detection accuracy. MPDIoU simplifies the process of calculating loss by using simpler distance calculations and weighting methods, eliminating the need to directly compute the intersection and union areas. This significantly reduces the computational load during training and improves processing speed, especially when dealing with a large number of bounding boxes.

To verify the effectiveness of the MPDIoU loss function during training, we conducted comparative experiments on training using CIoU and MPDIoU with the improved model. The experimental results are presented and analyzed in the ablation experiment section.

4 Experiment

4.1 Experimental Dataset

The dataset used in this experiment is the RESIDE RTTS dataset. RTTS is a comprehensive real-world dataset available under foggy conditions, containing 4322 natural foggy images with annotations for five object classes: person, bicycle, car, bus, and motorbike. We divided the RTTS dataset into a training set, validation set, and test set in a 7:2:1 ratio, resulting in 3025, 864, and 433 images, respectively. However, due to the relatively small number of images in the RTTS dataset, it is difficult to meet the large data requirements needed for deep learning model training. This limitation may affect the training efficacy and robustness of the model, making it challenging to achieve the desired training outcomes. To address this issue, this paper generates a foggy object detection dataset, VOC-Fog, on the public PASCAL VOC [33] dataset using an atmospheric scattering model to simulate traffic scenes with varying fog densities. The specific process is as follows: first, we select categories from VOC2007 and VOC2012 that match those in the RTTS dataset, totaling five categories: person, bicycle, car, bus, and motorbike. For images of these five categories, we randomly assign a value between 0.05 and 0.15 to the scattering coefficient, and a value between 0 and 1 to the atmospheric light value. These values are then substituted into the atmospheric scattering model formula to generate images with different fog densities and brightness levels. As a result, the synthetic foggy target detection dataset, VOC-Fog, contains a total of 16,441 images, with the training set, validation set, and test set comprising 10,870, 2776, and 2795 images, respectively. Finally, the proposed algorithm model is validated on both the RTTS dataset and the VOC-Fog dataset.

4.2 Experimental Metrics

This experiment utilizes the mAP to evaluate the detection performance of the model. mAP represents the mean precision obtained at different recall levels and is commonly used as the final metric for evaluating the performance of object detection algorithms. A great mAP value typically indicates that the model possesses superior detection performance. The detailed calculation is provided in Eqs. (14) to (17).

$$P_{\text{recision}} = \frac{Q_{\text{TP}}}{Q_{\text{TP}} + Q_{\text{FP}}} \tag{14}$$

$$R_{\text{ecall}} = \frac{Q_{\text{TP}}}{Q_{\text{TP}} + Q_{\text{FN}}} \tag{15}$$

$$AP = \int_0^1 PRdr \tag{16}$$

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{17}$$

In the above equations: Q_{TP} represents true positive, Q_{FP} represents false positive, and Q_{FN} represents false negative. R_{ecall} and P_{recision} represent the recall and precision, respectively. The AP value corresponds to the area enclosed by the PR curve with recall on the x-axis and precision on the y-axis. N denotes the number of categories.

4.3 Experimental Environment and Parameters

To ensure fairness and consistency in all experiments, the experimental parameters set for the two datasets are identical. The operating system used for the experiments is Windows 10 Professional, 64-bit, with 16 GB of RAM, an NVIDIA GeForce RTX 4060 GPU, and a 12th Gen Intel(R) Core (TM) i5-12400F 2.50 GHz processor. The parallel computing framework version is CUDA 12.3, and the deep learning framework used is PyTorch 2.1.0. The detailed training parameters are shown in the Table 2 below:

Table 2: All algorithm experimental parameters

Parameter names	Value
Training batchsize	8
Input image size	640 × 640 pixels
Epoch	175
Initial learning rate	0.01
Weight decay coefficient	0.0005
Momentum	0.937
Optimizer	SGD

4.4 Experimental Results and Analysis

4.4.1 Ablation Experiment

To verify the effectiveness and reliability of the proposed method, ablation experiments were conducted using the RTTS dataset and the VOC-Fog dataset to validate the effectiveness of each module and improvement.

As shown in Tables 3 and 4, the adoption of the Triplet Attention mechanism significantly enhances the network's ability to extract multi-scale and multi-dimensional features. On the RTTS dataset, the mAP50 increased by 1.3%, while on the VOC-Fog dataset, it improved by 1.2%. Similarly, the mAP50-95 saw an increase of 1.1% and 0.8%, respectively. These results indicate that the Triplet Attention mechanism effectively focuses the network on features across different scales, thereby improving detection performance in complex environments.

Table 3: Ablation experimental results of our model on the RTTS dataset

Triplet attention	DBB		MPDIoU	mAP%					mAP@50	mAP@50-95
	DetectEfficientHead	C2F-DBB		Person	Car	Bus	Bicycle	Motorbike		
				82.9	87.8	66.2	67.4	76.5	76.2	51.0
√				84.5	89.0	66.8	69.3	77.8	77.5	52.1
√	√			83.9	89.6	70.9	69.9	78.6	78.6	52.8
√	√	√		84.8	90.7	70.0	73.7	81.2	80.1	53.7
√	√	√	√	86.6	91.9	71.8	71.3	82.3	**80.8**	**54.0**

Table 4: Ablation experimental results of our model on the VOC-Fog dataset

Triplet attention	DBB		MPDIoU	mAP%					mAP@50	mAP@50-95
	DetectEfficientHead	C2F-DBB		Person	Car	Bus	Bicycle	Motorbike		
				81.9	81.0	76.9	75.6	74.0	77.9	57.5
√				82.3	82.7	77.0	76.2	77.1	79.1	58.3
√	√			83.4	83.2	78.3	77.6	78.5	80.2	59.0
√	√	√		84.1	84.9	79.1	78.5	79.4	81.2	59.6
√	√	√	√	84.5	85.5	79.7	78.9	79.8	**81.7**	**59.9**

With the introduction of our custom-designed DetectEfficientHead module, mAP50 increased by 1.1% on both datasets. This module enhances the network's ability to capture contextual information at the prediction layer by addressing the issue of parameter sharing between feature maps. As a result, it improves the detection ability of small objects and occluded objects.

After integrating the C2F-DBB module, mAP50 increased by 1.5% on the RTTS dataset and 1.0% on the VOC-Fog dataset, further demonstrating the importance of multi-scale feature fusion in the network's neck section. This module enhances the network's ability to utilize the relationships between feature maps by effectively fusing features at different scales, thereby significantly improving the overall performance of the network model.

After optimizing the loss function, mAP50 increased by 0.7% on the RTTS dataset and 0.5% on the VOC-Fog dataset. As shown in the Fig. 8, MPDIoU exhibits a faster convergence rate and consistently lower loss compared to CIOU. This indicates that MPDIoU outperforms CIOU in both the training and inference processes of object detection, effectively improving the model's convergence speed and inference capabilities.

In summary, the introduction and optimization of various modules have demonstrated significant performance improvements across different detection scenarios. Among them, the innovations in feature extraction and fusion provided by the Triplet Attention mechanism and the C2F-DBB module are particularly critical. The DetectEfficientHead module further enhances the model's robustness by capturing contextual information. Ultimately, the optimization of the MPDIoU loss function has effectively accelerated the training process and improved the overall performance of the model.

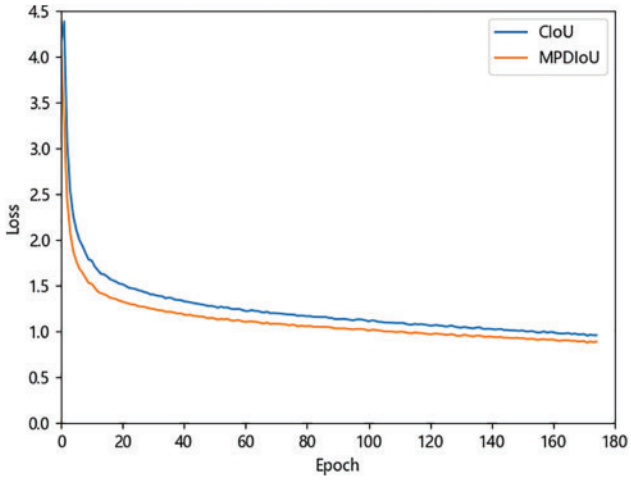


Figure 8: Loss function comparison diagram

4.4.2 Comparative Analysis of Experiments

To further verify the superior performance of the proposed improved algorithm in foggy scenarios and demonstrate that our model maintains high robustness even after defogging, we conducted defogging preprocessing on the RTTS and VOC-Fog datasets and used these defogged datasets for comparative experiments. Given that Light-DehazeNet [23] is a computationally efficient and lightweight convolutional neural network-based defogging algorithm, which also proposes a color visibility restoration method to avoid color distortion in defogged images, we chose to use Light-DehazeNet for defogging preprocessing of the RTTS and VOC-Fog datasets. The goal is to restore the image features to enhance the detection performance, robustness, and generalization ability of our model. The defogging effects are shown in Fig. 9.

Figure 9: Light-DehazeNet defogging effect in RTTS dataset and VOC-Fog dataset

We compared our model with YOLOv3, YOLOv5 [34], MS-DAYOLO [16], DSNet [3], IA-YOLO [15], DE-YOLO [35] and YOLOv9 [36]. Among these models, YOLOv3 and YOLOv5 are traditional series object detection models, and YOLOv9 represents the state-of-the-art detection

model. we combined them with the Light-DehazeNet algorithm for comparison. MS-DAYOLO, DSNet, IA-YOLO, DE-YOLO are currently mainstream detection algorithms for foggy scenarios, The comparative experimental results are presented in Tables 5 and 6, comparing the effectiveness of our proposed algorithm with other algorithms.

Table 5: Comparative experimental results of RTTS dataset after Light-DehazeNet (LDehaze) dehazing preprocessing

Detection algorithms	mAP%					mAP@50	mAP@50-95
	Person	Car	Bus	Bicycle	Motorbike		
YOLOv3	72.6	78.7	57.7	59.9	58.1	65.4	41.1
LDehaze-YOLOv3	73.4	79.3	58.2	60.6	58.7	66.0	41.5
YOLOv5s	80.6	86.3	64.0	66.5	66.5	72.8	48.5
LDehaze-YOLOv5s	81.1	86.7	64.8	67.3	67.4	73.5	49.0
MS-DAYOLO	39.8	43.5	32.1	31.4	31.8	35.7	20.6
DSNet	68.8	83.3	33.7	53.7	48.4	57.6	30.2
IA-YOLO	63.3	72.3	28.2	44.5	43.1	50.3	34.3
DE-YOLO	67.9	81.7	31.3	48.8	49.5	55.8	37.5
YOLOv9	86.5	89.3	72.1	71.0	72.2	78.2	51.2
LDehaze-YOLOv9	86.2	91.8	69.9	72.4	72.5	78.6	51.6
YOLOv8s	82.9	87.8	66.2	67.4	76.5	76.2	51.0
LDehaze-YOLOv8s	83.6	88.4	66.9	68.1	77.0	76.8	51.6
Ours	86.6	91.9	71.8	71.3	82.3	**80.8**	**54.0**
LDehaze-Ours	87.9	92.3	72.7	72.2	82.3	**81.5**	**54.4**

Table 6: Comparative experimental results of VOC-Fog dataset after Light-DehazeNet dehazing preprocessing

Detection algorithms	mAP%					mAP@50	mAP@50-95
	Person	Car	Bus	Bicycle	Motorbike		
YOLOv3	74.7	72.4	65.0	68.2	68.1	69.7	45.1
LDehaze-YOLOv3	75.1	73.1	65.4	68.5	68.7	70.2	45.3
YOLOv5s	81.6	81.1	75.8	75.3	75.0	77.8	56.2
LDehaze-YOLOv5s	82.2	81.8	76.6	75.9	75.8	78.5	56.7
MS-DAYOLO	66.9	66.4	61.7	57.8	60.6	63.0	36.4
DSNet	71.5	70.3	84.2	68.6	62.3	71.4	40.7
IA-YOLO	70.3	71.9	77.1	68.5	61.8	69.9	45.4
DE-YOLO	76.0	74.9	75.2	71.6	68.2	73.2	48.7
YOLOv9	84.9	82.7	77.1	78.0	75.6	79.7	52.1
LDehaze-YOLOv9	83.8	83.4	78.2	77.5	77.4	80.1	52.5
YOLOv8s	81.9	81.0	76.9	75.6	74.0	77.9	57.5

(Continued)

Table 6 (continued)

Detection algorithms	mAP%					mAP@50	mAP@50-95
	Person	Car	Bus	Bicycle	Motorbike		
LDehaze-YOLOv8s	82.5	81.7	77.3	76.0	74.3	78.4	57.7
Ours	84.5	85.5	79.7	78.9	79.8	**81.7**	**59.9**
LDehaze-Ours	85.1	86.0	80.2	79.7	80.4	**82.3**	**60.2**

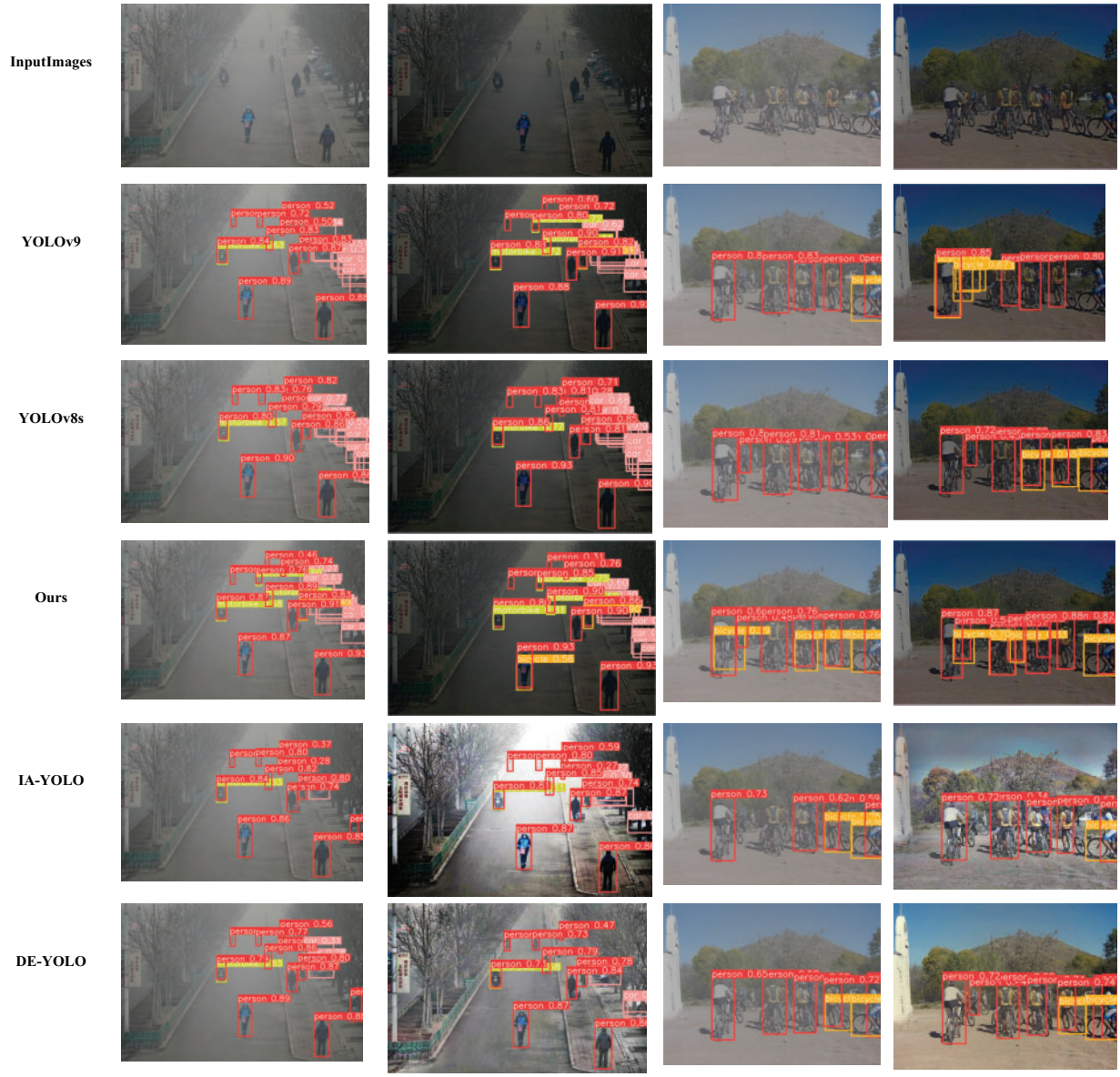
The Tables 5 and 6 show that our model outperforms the other models in terms of average mAP performance across both datasets. Specifically, our method achieves an mAP50 of 80.8% on the RTTS dataset, representing a 4.6% improvement over the baseline model. After defogging preprocessing, this increases to 81.5%, a further 0.7% improvement. On the VOC-Fog dataset, our method achieves an mAP50 of 81.7%, a 3.8% increase over the baseline model, with defogging preprocessing, this increases to 82.3%, an additional 0.6% improvement. We also observed that traditional YOLO series algorithms showed mAP improvements after defogging preprocessing, demonstrating that using the Light-DehazeNet defogging algorithm significantly enhances detection accuracy. This is because the defogged images exhibit higher contrast and more pronounced edge information, making the feature information of the original image content clearer and thereby improving detection performance. Furthermore, the tables reveal that the average performance on the VOC-Fog dataset is consistently higher than on the RTTS dataset. This discrepancy is attributed to the fact that the VOC-Fog dataset is artificially synthesized, and the complexity of haze images in this dataset is significantly lower than that of real-world haze images. Through comparative experiments, our model demonstrates a significant performance advantage over other object detection models, validating that our method enhances detection accuracy in foggy scenarios and meets the requirements of related detection tasks.

4.4.3 Comparative Analysis of Detection Effects

To clearly demonstrate superior detection performance, we compared our method with the baseline model YOLOv8, the latest detection algorithm YOLOv9, and two mainstream foggy scene object detection methods from the past two years, IA-YOLO and DE-YOLO, as shown in Fig. 10. To ensure consistency and fairness in the experiments, we used the same image from the same dataset for comparison, with dehazing preprocessing applied. Columns (a) and (b) represent the detection results before and after dehazing preprocessing on the RTTS dataset, while Columns (c) and (d) display the results on the VOC-Fog dataset. In the IA-YOLO and DE-YOLO model, since both methods incorporate dehazing enhancement modules, when we did the non-defogging experiments, we removed these modules prior to conducting experiments to ensure consistency and rationality in our comparative experiments, highlighting the superiority of our method.

By comparing the results before and after the algorithm improvement, as well as the effects before dehazing, we observe a vertical comparison in Columns (a) and (c) of Fig. 10. For instance, in the RTTS dataset, our method successfully detects small targets such as pedestrians behind the fog, motorcycles, and bicycles, which other models fail to identify. In the VOC-Fog dataset, our method detects more bicycles, whereas other models miss some of these targets to varying degrees. Our method effectively detects pedestrians and vehicles obscured by fog. Compared to the baseline model and other competitors, the improved algorithm in this paper significantly reduces the rates of missed and false

detections while improving detection accuracy. This demonstrates the effectiveness of the proposed algorithm improvements.



(a) Before dehazing on RTTS (b) After dehazing on RTTS (c) Before dehazing on VOC-Fog (d) After dehazing on VOC-Fog

Figure 10: Comparison of detection results between our method and several competitors on the RTTS and VOC-Fog datasets. Columns (a) and (b) represent the detection results before and after dehazing preprocessing on the RTTS dataset, Columns (c) and (d) display the results on the VOC-Fog dataset

By comparing the detection results before and after dehazing, specifically between Columns (a) and (b), and between Columns (c) and (d), it is evident that all methods exhibit some level of improvement after dehazing. Among them, our method is the most obvious and more advantageous.

Compared to other models, our method significantly reduces the rate of missed detections and further increases the detection accuracy percentage. It also effectively detects occluded targets. However, due to the introduction of light attenuation in real foggy images and images synthesized using the atmospheric scattering model, During the dehazing process, the removal of certain factors may cause the images to darken, potentially leading to the loss of some feature information. However, our method, when combined with the advantages of the Light-DehazeNet algorithm, allows for better preservation of feature information, making the overall features more prominent. Experimental results demonstrate that the dehazing algorithm has a positive impact on improving object detection performance.

Based on the above data and detection results, the proposed algorithm significantly reduces the miss detection rate and false detection rate in foggy scenarios. The detection accuracy is markedly improved, meeting the detection task requirements in foggy traffic scenarios.

5 Conclusion

To improve the accuracy and robustness of object detection algorithm in foggy traffic scenarios, this paper proposes a multi-scale object detection algorithm based on YOLOv8. A lightweight attention mechanism, Triplet Attention, is introduced between the backbone and neck parts of the network to further extract feature information. In the neck part, the DBB branch module is incorporated into the C2F module to enhance multi-scale fusion of feature information at different levels. Additionally, based on the DBB module, we redesigned the original network's detection head and proposed a new decoupling head, DetectEfficientHead, to improve detection performance and reduce missed and false detections. Regarding the loss function, we adopted MPDIoU, which, in addition to considering factors such as centroid distance, width, and height deviations, also takes into account the minimum point distance. Compared to the original CIOU loss function, MPDIoU accelerates the convergence speed of the loss during training and optimizes the network's inference capability. Finally, experiments were conducted on the RTTS and VOC-Fog datasets, and the results showed that the improved method proposed in this paper achieved the highest accuracy and demonstrated great performance, proving the effectiveness and practicality of the proposed method.

In future work, this research will use distillation and pruning techniques to reduce unnecessary parameters and redundancy in the model while maintaining accuracy. This will enhance the real-time performance of the detection tasks and improve the model's adaptability and generalization capability in various adverse weather conditions.

Acknowledgement: The authors would like to thank the anonymous reviewers and the editor for their valuable suggestions, which greatly contributed to the improved quality of this article.

Funding Statement: This work was supported by the National Natural Science Foundation of China (Grant Nos. 62101275 and 62101274).

Author Contributions: Study conception and design: Honglin Wang, Zitong Shi. Data collection, analysis and interpretation of results: Zitong Shi, Cheng Zhu. Draft manuscript preparation: Honglin Wang, Zitong Shi. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data used to support the findings of this study are available from the corresponding author upon request.

Ethics Approval: Not applicable.

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

References

[1] B. Li, R. T. T. Tan, D. P. Mandic, H. Zhang, W. Ren and Z. Wang, "AOD-Net: All-in-one dehazing network," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 4770–4778.

[2] 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. 2016. doi: 10.1109/TPAMI.2016.2577031.

[3] C. Huang, T. -H. Le, and D. -W. Jaw, "DSNet: Joint semantic learning for object detection in inclement weather conditions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 8, pp. 2623–2633, Aug. 2020. doi: 10.1109/TPAMI.2020.2977911.

[4] B. Li *et al.*, "Benchmarking single-image dehazing and beyond," *IEEE Trans. Image Process.*, vol. 28, no. 1, pp. 492–505, Jan. 2018. doi: 10.1109/TIP.2018.2867951.

[5] R. Girshick *et al.*, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587. doi: 10.1109/CVPR.2014.81.

[6] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vision (ICCV)*, Santiago, Chile, 2015, pp. 1440–1448. doi: 10.1109/ICCV.2015.169.

[7] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vision (ICCV)*, Venice, Italy, 2017, pp. 2961–2969. doi: 10.1109/ICCV.2017.322.

[8] M. Hu, Y. Zhang, Y. Li, Z. Zhao, and L. Wang, "DAGL-Faster: Domain adaptive faster R-CNN for vehicle object detection in rainy and foggy weather conditions," *Displays*, vol. 79, no. 23, Aug. 2023, Art. no. 102484. doi: 10.1016/j.displa.2023.102484.

[9] Y. Chen *et al.*, "Domain adaptive faster R-CNN for object detection in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3339–3348. doi: 10.1109/CVPR.2018.00352.

[10] C. Ge *et al.*, "Rethinking attentive object detection via neural attention learning," *IEEE Trans. Image Process*, vol. 33, pp. 1726–1739, Jul. 2023. doi: 10.1109/TIP.2023.3251693.

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 779–788. doi: 10.1109/CVPR.2016.91.

[12] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands, Oct. 2016, pp. 21–37. doi: 10.1007/978-3-319-46448-0_2.

[13] Q. H. Chen, Y. Yang, J. Wang, and L. Zhang, "Vehicle and pedestrian detection based on AOD-Net and SSD algorithm in hazy environment," (in Chinese), *J. Chongqing Univ. Technol. (Nat. Sci.)*, vol. 35, no. 5, pp. 108–117, May 2021.

[14] S. Liu, X. Zhang, L. Wu, and Y. Li, "Improved object detection of YOLOv4 in foggy conditions," *J. Syst. Simul.*, vol. 35, no. 8, pp. 1681–1691, Aug. 2023.

[15] W. Liu, L. Zhang, Y. Zhao, and Z. Li, "Image-adaptive YOLO for object detection in adverse weather conditions," *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 2, pp. 1562–1570, Jul. 2022. doi: 10.1609/aaai.v36i2.20072.

[16] M. Hnewa and H. Radha, "Multiscale domain adaptive YOLO for cross-domain object detection," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2021, pp. 3323–3327. doi: 10.1109/ICIP42928.2021.9506039.

[17] Y. Guo, "Based on section local histogram equalization image enhancement algorithm," in *Proc. 14th Int. Conf. Machine Learn. Comput.*, 2022, pp. 461–465. doi: 10.1145/3529836.3529946.

[18] Z. Wei, X. Wang, J. Zhang, and L. Zhang, "An image fusion dehazing algorithm based on dark channel prior and Retinex," *Int. J. Comput. Sci. Eng.*, vol. 23, no. 2, pp. 115–123, Feb. 2020. doi: 10.1504/IJCSE.2020.110556.

[19] M. Ju, X. Liu, M. Li, and C. Li, "BDPK: Bayesian dehazing using prior knowledge," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 8, pp. 2349–2362, Aug. 2018. doi: 10.1109/TCSVT.2018.2869594.

[20] S. G. Narasimhan and S. K. Nayar, "Contrast restoration of weather degraded images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 6, pp. 713–724, Jun. 2003. doi: 10.1109/TPAMI.2003.1201821.

[21] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2341–2353, Dec. 2010. doi: 10.1109/TPAMI.2010.168.
[22] B. Cai, K. Wang, and W. Shen, "DehazeNet: An end-to-end system for single image haze removal," *IEEE Trans. Image Process.*, vol. 25, no. 11, pp. 5187–5198, Nov. 2016. doi: 10.1109/TIP.2016.2598681.
[23] H. Ullah, A. K. M. Mahedi, M. A. K. Azad, and K. M. M. Rahman, "Light-DehazeNet: A novel lightweight CNN architecture for single image dehazing," *IEEE Trans. Image Process.*, vol. 30, pp. 8968–8982, Dec. 2021. doi: 10.1109/TIP.2021.3116790.
[24] D. Misra, T. Nalamada, A. U. Arasanipalai, and Q. Hou, "Rotate to attend: Convolutional triplet attention module," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, Jan. 2021, pp. 3139–3148.
[25] X. Ding, X. Zhang, J. Han, and G. Ding, "Diverse branch block: Building a convolution as an inception-like unit," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit*, Jun. 2021, pp. 10886–10895. doi: 10.1109/CVPR46437.2021.01085.
[26] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye and D. Ren, "Distance-IoU loss: Faster and better learning for bounding box regression," *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 7, pp. 12993–13000, Apr. 2020. doi: 10.1609/aaai.v34i07.6999.
[27] S. Ma and Y. Xu, "MPDIoU: A loss for efficient and accurate bounding box regression," Jul. 2023, *arXiv:2307.07662*.
[28] S. Woo *et al.*, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 3–19. doi: 10.1007/978-3-030-01234-2_1.
[29] W. Chen *et al.*, "Multi-attention network for compressed video referring object segmentation," in *Proc. 30th ACM Int. Conf. Multimed.*, 2022, pp. 4416–4425. doi: 10.1145/3503161.3547761.
[30] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141. doi: 10.1109/CVPR.2018.00745.
[31] J. Park, S. Woo, J. Y. Lee, and I. S. Kweon, "BAM: Bottleneck attention module," Jul. 2018, *arXiv:1807.06514*.
[32] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "NGCNet: On-local networks meet squeeze-excitation networks and beyond," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops*, Oct. 2019, pp. 1971–1980. doi: 10.1109/ICCVW.2019.00246.
[33] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010. doi: 10.1007/s11263-009-0275-4.
[34] G. Jocher, A. Chaurasia, J. Qiu, and N. Stoken, "Ultralytics/yolov5: v5.0-YOLOv5-P6 1280 models, AWS, Supervise.ly and YouTube integrations," *Zenodo*, 2021. doi: 10.5281/zenodo.4679653.
[35] Q. Qin *et al.*, "DENet: Detection-driven enhancement network for object detection under adverse weather conditions," in *Proc. Asian Conf. Comput. Vis.*, Dec. 2022, pp. 2813–2829. doi: 10.1007/978-3-031-26313-2_30.
[36] C. -Y. Wang, I. -H. Yeh, and H. -Y. M. Liao, "YOLOv9: Learning what you want to learn using programmable gradient information," 2024, *arXiv:2402.13616*.