IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 10 July 2024, accepted 31 July 2024, date of publication 5 August 2024, date of current version 13 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3438612

RESEARCH ARTICLE

Road Object Detection in Foggy Complex Scenes Based on Improved YOLOv8

LONG CHENG, DAN ZHANG, AND YAN ZHENG
School of Automotive and Traffic Engineering, Jiangsu University of Technology, Changzhou, Jiangsu 213001, China
Corresponding author: Yan Zheng (zhengyan@jsut.edu.cn)

This work was supported by the Science and Technology Project of Changzhou under Grant CE20220066 and Grant CE20230021.

ABSTRACT Focusing on the challenges of vehicle detection in foggy weather, especially the algorithm of low accuracy caused by small and incomplete targets in adverse weather conditions, a foggy weather vehicle detection algorithm based on improved lightweight YOLOv8 was proposed. Firstly, the dataset was processed through a combination of data transformation, Dehaze Formers and dark channel preprocessing. Secondly, in the main body of YOLOv8, the C2f component was replaced with the dynamic convolution C2f- DCN, enhancing its adaptability to geometric changes in the image. To further improve the detection performance of the classifier, an improved S5attention module based on S2-MLP was introduced. This module utilizes contextual information to capture long-range dependencies and assign weights to different channels based on their relevance to the task at hand. By considering non-local features, the S5attention module helps the model better capture important spatial relationships within the image. Additionally, the feature extraction module was updated to FasterNext, improving the differential convolution's feature extraction capabilities. The Involution module was also introduced to reduce FLOPs during feature channel fusion and reduce the model's parameter count. Experimental results show that on the RESIDE foggy weather dataset, the improved algorithm has an mAP50 increase of 4.1% compared with the original algorithm, and the model's parameter quantity is only 9.06m, with a computational cost reduced from 28.7G to 28.1G. The research model in this article will provide technical support for detecting vehicle targets in foggy weather, ensuring fast and accurate operation.

INDEX TERMS Deep learning, foggy weather vehicle detection, YOLOv8, feature extraction.

I. INTRODUCTION

Complex weather condition is one of an important cause of traffic accidents. Extreme weather conditions such as fog, rain, and snow greatly reduce the visibility of roads and make driving extremely dangerous. In these adverse weather conditions, drivers often find it difficult to detect obstacles and other vehicles in front of them in a timely manner, resulting in collisions and traffic chaos. Solving traffic safety issues in complex weather conditions is crucial, and timely detection and prevention of vehicles and obstacles is of great significance for maintaining traffic safety.

Vehicle detection technology in foggy weather mainly falls into two categories: traditional object detection algorithms and object detection algorithms based on deep learning. Traditional foggy road object detection methods can be divided into two major approaches. One approach involves a two-stage method where the first stage involves pre-processing the image to remove fog, and the second stage involves feeding the processed image into an object detection model for detection. Li et al. [1] combined the PDR-Net defogging network with Faster RCNN, significantly improving the network's ability to understand image information, especially under foggy conditions. However, this method introduces artifacts in the processed image, which to some extent affects the image quality and detection accuracy. The other approach adopts a one-stage strategy that integrates defogging and detection, such as Xiaomin et al. [2] proposed end-to-end adaptive defogging generation network. Through a clever two-stage mapping strategy, the defogging output

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

of the primary network is used as input for the secondary network, optimizing the defogging effect. Nonetheless, due to its limited reliance on prior information and insufficient utilization of scene depth information, the defogging effect on distant targets is not satisfactory. In summary, defogging algorithms still face issues such as loss of target information and image blurring. Another category of deep learning-based methods, through multiple layers of convolution and pooling operations, can automatically learn patterns and structures in images. With their high detection accuracy, strong generalization performance, and other advantages, algorithms can more accurately locate and recognize targets in various complex backgrounds.

Numerous CNN-based object detection models have been introduced recently, and they can be broadly categorized into two types: one-stage and two-stage detectors. The two-stage algorithms include R-CNN [3] and Faster-RCNN [4], among others [5], [6]. These algorithms first identify potential object regions and then classify them. However, their two-stage nature can limit their efficiency in practical applications. If it is directly used for road target detection, it is difficult to meet the real-time requirements. In contrast, one-stage object detectors directly produce localization and classification from dense predictions derived from feature maps. This approach offers superior speed and is well-suited for scenarios with real-time requirements. One-stage object detection algorithms include YOLO series algorithms (you only look once) [7], [8], [9], [10], [11], [12], SSD algorithms (Single Shot MultiBox Detector) [13], and so on. In this context, numerous studies have been dedicated to refining single-stage object detection algorithms, aiming to enhance their practical utility and efficiency. For instance, Gao et al. [14] improved the single-stage detection algorithm for traffic sign detection based on SSD by incorporating depth wise separable convolution to enhance feature extraction. However, due to the SSD algorithm's tendency to have a high Intersection over Union (IOU) for small-sized objects on lower-level feature maps, the algorithm performs poorly in handling small-sized target. Xuan et al. [15] proposed a target detection algorithm for traffic scenes under complex meteorological conditions. They introduced DenseNet and dilated convolution to improve the YOLOv3 structure, which had a good detection effect on images taken under complex meteorological conditions. However, because it used a dark channel defogging algorithm to enhance the image, the effect of image processing containing sky areas was poor, this limits the algorithm's generalization ability in various complex scenarios. Wang et al. [16] artificially generated fog images through an atmospheric scattering model and the depth information of images to expand the sample size. However, they did not take into account the differences that exist in actual foggy scenes, which may adversely affect the generalization performance of the model. Ze et al. [17] proposed CSPDarkNet-53 as the backbone network for feature extraction from low-illuminance images. Additionally, they introduced the Path Aggregation Enhancement Module (PAEM) to further enhance the representation capability of these features. This approach effectively addressed common issues in low-illuminance images, such as low brightness, excessive noise, and loss of detailed information. However, this improvement also brought an increase in computational complexity, which subsequently led to a decrease in detection speed. Yin et al. [18] optimized the structure of YOLOv5 to address the issue of low recognition accuracy of traffic annotations in haze weather and use the K-means clustering algorithm to re-cluster the anchor boxes. They reduced the depth of the feature pyramid and limited the maximum down sampling ratio. However, a deeper feature pyramid helps capture multi-scale information, and reducing its depth may compromise the detection performance for small targets. Kai et al. [19] combined the ideas of feature separation and merging, introduced the SPPCSPC module, and utilized coordinated attention from the efficient mobile network design (CA) module to enhance the detection capability of YOLOv7 in small target scenarios. Despite significant progress made in enhancing target detection performance under complex weather conditions, numerous challenges and limitations remain. In particular, for small object detection, current methods are constrained by environmental disturbances and image blurriness, making it difficult to accurately extract and detect features, resulting in suboptimal detection outcomes. Moreover, inadequate generalization of image enhancement techniques, a lack of diversity in sample generation strategies, and the trade-off between computational complexity and detection speed are crucial factors limiting further improvements in detection performance. Notably, while YOLO series algorithms excel at detecting objects of all sizes, their performance for specific-sized objects, particularly in complex scenarios such as foggy weather, often lags behind dedicated small object detection algorithms [20], [21]. Building upon this foundation, this paper aims to explore and optimize the detection capabilities of YOLO series algorithms for special-sized objects under complex weather conditions, particularly in foggy scenarios, by proposing an efficient and accurate detection algorithm that provides robust technical support for fields like intelligent transportation and autonomous driving.

This study introduces a road target detection algorithm tailored for foggy weather, building upon an enhanced YOLOv8 model. The advancements made in this algorithm are manifested in the following steps:

1) We replaced the C2f layer in the backbone network with the C2f-DCN module, which introduces a deformable convolution kernel. By dynamically adjusting the shape and position of the convolution kernel, it can more effectively capture the features of objects with blurred contours and uneven scales in foggy conditions.

2) The Involution module and FasterNet module are applied to the feature extraction layer. The Involution module utilizes its spatial specificity to extract spatial contextual

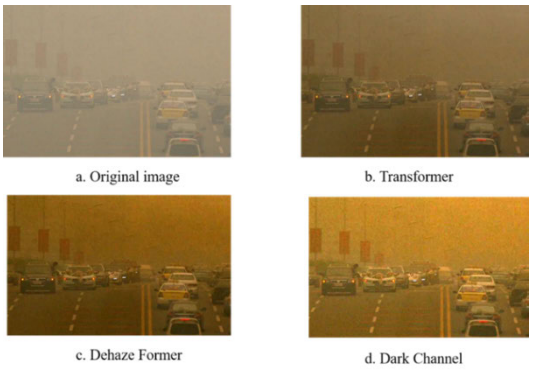

FIGURE 1. Examples of raw image and data enhancement image.

information and adaptively assign weight files, thereby improving the model's ability to extract small targets and blurred images. The FasterNet module adopts a partial convolution (PConv) strategy, effectively reducing computational complexity and memory access, and improving the network's inference speed and computational efficiency.

3) We propose an improved S5-attention module based on S2-MLP. This module captures non-local features in the image through clever spatial displacement operations by fusing feature maps of different scales, better addressing the challenges posed by occlusion and small objects.

This paper is divided into the following sections: Section II introduces the selected dataset and the improved methodology presented in this paper; Section III focuses on experimental results and comparative experiments; Section IV provides conclusions as well as directions for subsequent work and improvements.

II. MATERIALS AND METHODS

A. DATA COLLECTION AND PRETREATMENT

In this experiment, a new large-scale benchmark dataset consisting of both synthetic and real-world blurred images, referred to as RESIDE (Realistic Single Image Dehazing) [22] was utilized. Considering the complexity of foggy scenes, three advanced data augmentation methods involving Transformer, Dehaze Formers, and Dark Channel are adopted to prevent overfitting during training. These methods not only help generate richer and more diverse training data sets, but also significantly improve the model's generalization ability. From this dataset, 16,000 foggy images were selected and divided into training, testing, and validation sets in an 8:1:1 ratio. Additionally, data augmentation was performed on 2,000 of these images, resulting in 6,000 augmented images. These images primarily encompass five major human vehicle scene categories: Person, Car, Bus, Bicycle, and Motorbike. Figure1 presents typical images before and after data augmentation.

B. IMPROVED YOLOv8 MODEL

Real-time object detection has become a critical component in numerous applications, including autonomous driving, robotics, and unmanned aerial vehicles (UAVs). Among the various object detection algorithms, the YOLO algorithm stands out for its speed and accuracy. With the release of the YOLOv8 version, the algorithm not only meets real-time requirements but also achieves faster and more accurate results compared to previous versions, while minimizing computational complexity, parameter count, and model complexity. This makes it suitable for targeted optimization and modification of objects in autonomous driving scenarios. The backbone network consists of Conv, C2f, and SPPF [23] modules. The Conv module performs convolution, Batch Normalization (BN) [24], and SILU activation function operations on the input image. The C2f module introduces modifications to the CSPLayer, incorporating a cross-stage partial bottleneck with two convolutions that combines deep features with contextual information, enhancing inference speed and detection accuracy. The SPPF module is a spatial pyramid pooling layer inspired by SPP, which addresses redundant feature information ex-traction in convolutional neural networks, enabling local and global feature fusion and enriching feature information. In the head segment, a popular decoupled head structure is implemented, which separates the classification head from the detection head. Compared to other algorithms, YOLOv8 is extremely friendly in practical deployment, boasting high accuracy while consuming significantly fewer resources than the transformer [25] structure. This makes it capable of running smoothly on various hardware platforms. This optimized balance between high performance and resource consumption makes YOLOv8 a highly competitive object detection algorithm in practical applications. However, the original YOLOv8 model demonstrates less than ideal performance when dealing with small objects. This phenomenon is primarily attributed to its task-aligned assignment mechanism, which relies on the model's prediction score and Intersection over Union (IOU). Due to the uneven distribution of large and small targets in the dataset, the label assignment for positive and negative samples is not accurate enough in the initial stages of model training, which can adversely affect the convergence of results. Secondly, slight changes in the position of small targets can cause fluctuations in IOU, affecting the localization of these targets. To enhance the accuracy of vehicle detection in foggy weather, an improved detection network model, YOLOv8-DF, based on YOLOv8s is proposed. The network structure of the algorithm model YOLOv8-DF presented in this paper is depicted in Figure 2, which high-lights three main improvements. To address the issue of small and incomplete vehicle targets in complex weather conditions, the cf2-DCN module is introduced, which incorporates offset weights and positional information. To reduce the number of parameters and enhance model lightweighting, the involution model is introduced. The FasterNet module strengthens information processing capabilities, accelerates the fusion of network feature information, and improves prediction accuracy. The S5Attention serves as a crucial attention mechanism, strengthening the meticulous capture

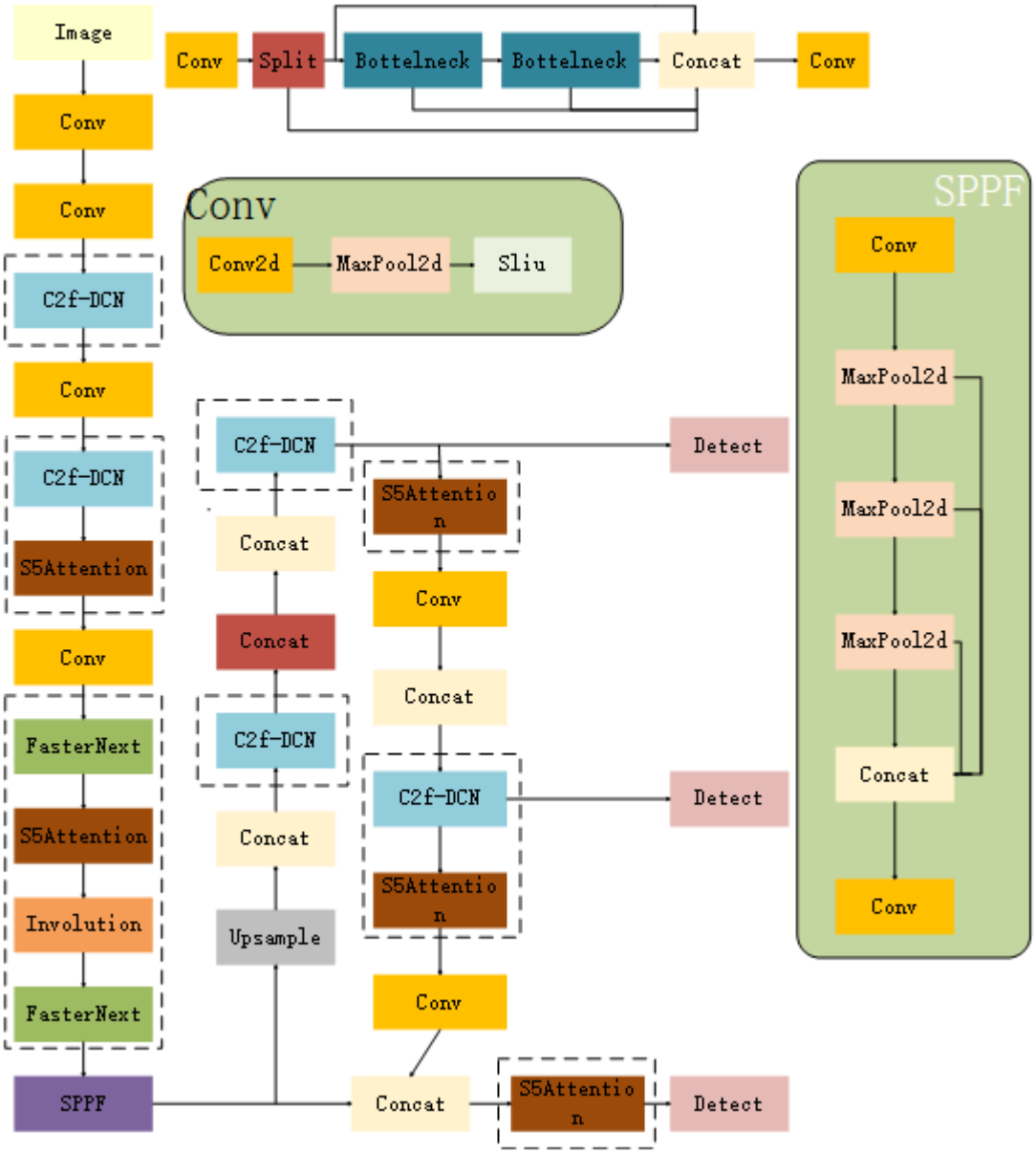

FIGURE 2. Structure diagram of the improved model of YOLOV8-DF.

of image details, simplifying the architecture, and boosting computational efficiency.

C. DEFORMABLE CONVOLUTION

Due to the complexities of road scene targets, particularly in foggy weather, there are small target sizes and incomplete target information. The original YOLOv8 model exhibited poor performance in detecting small targets after feature extraction by the backbone network. To effectively handle and represent multi-scale features and improve the modeling ability for deformed targets, the C2f module was modified to add Deformable Conv [26], [27] to correct the amplitude of input features from different spatial locations. This subtly design combines offset weights and positional information, enabling the network to better adapt to objects of varying scales and shapes. The module increased the target detection size range of the YOLOv8 network and enhanced its detection robustness in complex road backgrounds. Figure 3 demonstrates the sampling methods of regular convolution and deformable convolution with kernel size of 3×3. It can be divided into two steps: (1) By adding a displacement to the traditional convolution process, we predict the convolution offset from the input feature map. (2) We set a penalty term coefficient based on the convolution offset position to prevent the convolution offset from exceeding a certain range and optimize the sampling region. This coefficient not only limits the range of convolution offset, but also helps optimize

the sampling area, thereby improving the accuracy and effectiveness of feature extraction. The calculation process for the output feature map y at position P is as follows:

$$y(P) = \sum_{k=1}^{K} W_k * x\left(P + P_n + \Delta P_n\right) * \Delta m_k \tag{1}$$

Traditional convolution computes the output feature map by performing a weighted sum over each point in the input feature map, using the formula ppp. Here, each output point y(p) is obtained by weighted summation of the input x at position P+Pn with the convolution kernel Wk, aligning with the center of the kernel. However, in Deformable Convolution (DCONV), an offset Pn is introduced to enhance the model's geometric adaptability. This offset allows the convolution kernel to sample at non-fixed positions on the input feature map. To prevent unreasonable deformations from negatively impacting the model's ability to learn image information, DCONV also incorporates a weighting coefficient mk. This coefficient adjusts the effectiveness of the sampling region determined by the offset, ensuring that the model focuses on truly meaningful image areas and thus improving the accuracy and efficiency of feature extraction.

D. LIGHTWEIGHT BACKBONE NETWORK

In order to improve network inference speed and reduce network parameters and computational cost, convolutional modules are replaced in the backbone network with involution. Classical convolution ensures spatial invariance; i.e., the sharing of information by the convolution kernel at different locations, but its locality limits the high-level receptive field, making feature extraction for small targets and blurry images difficult. However, involution can extract feature information from a broader spatial context and adaptively allocate weight coefficients according to spatial differences, prioritizing the most important information. Involution has fewer parameters and reduces model complexity. Figure 4 shows the specific process of involution.

For input and output feature maps, the input is denoted as $F \in \mathbb{R}^{C_{\text{in}} \times \text{H} \times \text{W}}$ and the output as $F_{\text{out}} \in \mathbb{R}^{C_{\text{out}} \times \text{H} \times \text{W}}$. Where C_{in} represents the number of channels in the input feature map, C_{out} represents the number of channels in the output feature map, and H, W correspond to the height and width of the feature map, respectively. Given the channel specificity of convolutional operations, the C_{out} groups of convolution kernels can be represented as $C \in \mathbb{R}^{C_{in} \times C_{\text{out}} \times \text{K} \times \text{K}}$, where K denotes the size of the convolution kernel. After each group of convolution kernels processes the input feature map, it generates a corresponding output $F'_c \in \mathbb{R}^{\text{H} \times \text{W}}$, where $\text{c} = 1, 2, C_{\text{in}}$. Finally, all c are integrated to obtain the final input feature map $F_{\text{out}} \in \mathbb{R}^{C_{\text{out}} \times H \times W}$.

Contrary to the design principle of traditional convolution, Involution emphasizes spatial specificity when processing input feature maps, meaning that different convolution kernels are used in different spaces within the same group. The process of generating the Involution kernel can be represented as:

$$I_{i,j} = \phi\left(F_{i,j}\right) = W_1 \sigma\left(W_0 F_{i,j}\right) \tag{2}$$

In the formula, $\phi(\bullet)$ is the generation function of the Involution kernel, which consists of, $W_0 \in R_r^{\frac{c}{c}}$, $W_1 \in R^{K*K*G*C/r}$, where (r) represents the scaling ratio; $\sigma = \text{Relu}(BN(\bullet))$ is the intermediate batch normalization (BN) and ReLU function. By selecting a feature vector $F_{i,j} \in \mathbb{R}^{1\times1\times C}$ from a certain pixel on the input feature map (F), a new feature vector $F'_{i,j} \in \mathbb{R}^{1\times1\times\text{C}}$ is obtained through $\phi\left(W_0 - F_{BN} - F_{RELU} - W_1\right)$. This new feature vector is then reshaped to obtain the Involution kernel for that pixel. Finally, the output feature map F_{out} is obtained by multiplying and adding the feature vectors of adjacent coordinates. When optimizing deep learning networks, we often face the challenge of striking a balance between reducing network parameters and memory access. Although operators can effectively decrease the number of parameters, they may increase memory access due to additional data processing steps such as concatenation, data rearrangement, and pooling, which are crucial for enhancing network inference speed. Regarding the V8 model, we observed redundant computations in its network structure, which not only increased floating point operations (FLOPs) but also led to increased model processing latency. Model latency can be described by the formula $\text{Latency} = \frac{FLOPs}{FLOPS}$. Therefore, the improvement focuses on how to effectively reduce FLOPs and increase floating point operations per second (FLOPS) while maintaining accuracy, to achieve the goal of reducing latency and improving overall computation speed. Depthwise separable convolution [28] (DWConv) significantly reduces redundant computations and FLOPs by combining depthwise convolution and pointwise convolution. However, it's worth noting that since it only operates on a single channel during the separated convolution stage, ignoring the correlation between channels, directly replacing regular convolution may lead to a decrease in model accuracy. To compensate for this accuracy loss, the method of increasing the number of DWConv channels from c to c' was adopted to improve model accuracy. Nevertheless, this approach also increases the computational burden and memory access cost accordingly. The memory access amount of DWConv can be represented by formula (3), where h and w represent the length and width of the feature map, respectively, and c represents the number of channels.

$$h \times w \times 2c' + k^2 \times c' \approx h \times w \times 2c' \tag{3}$$

The memory accesses for regular convolution are:

$$h \times w \times 2C + k^2 \times C \approx h \times w \times 2C \tag{4}$$

When $\text{c}' > \text{c}$, the memory access amount of DWConv will be higher than that of regular convolution. Therefore, a new type of convolution module is needed to address the efficiency deficiencies of both regular convolution and

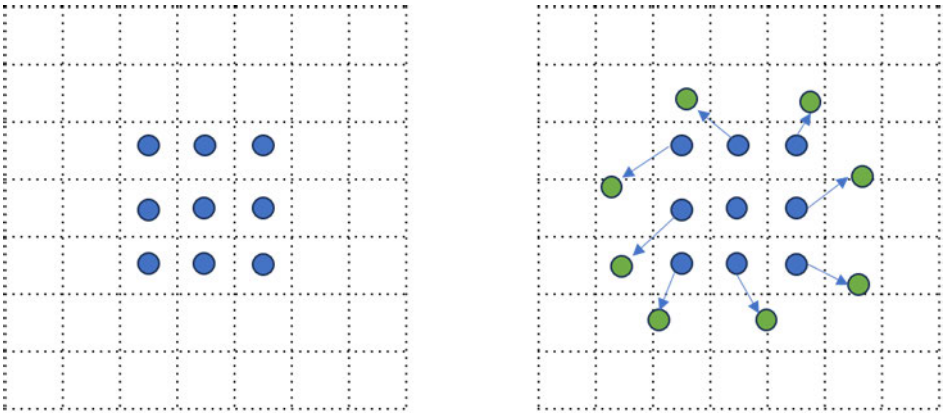

FIGURE 3. Convolution and deformable convolution process: a. Convolution, b. Deformable convolution The blue arrow represents the displacement amount added to the sampling.

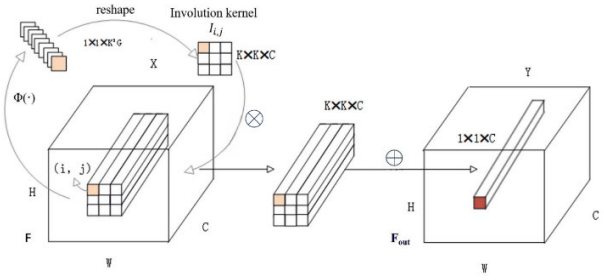

FIGURE 4. The specific process of involution.

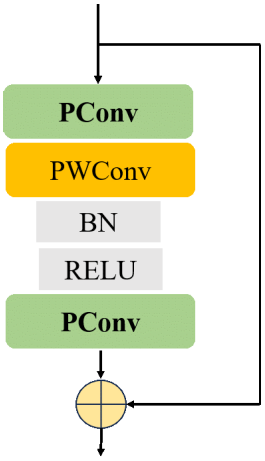

FIGURE 5. Structure of FasterNet block.

DWConv, thereby improving detection speed. In the FasterNet [29] architecture, an improved method for traditional convolution is adopted: only a portion of the input channels undergo regular convolution for feature extraction, while the remaining channels remain unchanged. Compared to regular convolution, PConv (Partial Convolution) exhibits higher efficiency because it only processes data from a subset of channels. Specifically, when the partial ratio is set to 1/4 (i.e., r = cp/c = 1/4, where cp represents the number of affected channels and c represents the total number of channels), only 1/4 of the channels undergo convolution calculations. This optimization significantly reduces computational complexity, making the computational complexity of PConv [30] only 1/16 that of regular convolution. The FasterNet module consists of two PWCONV layers and one PCONV layer. BN and RELU are added in the next two PWCONV layers to accelerate model training and avoid gradient vanishing problems, effectively reducing latency and maintaining efficient flow of feature information. Its structure is shown in the figure 5.

E. ENHANCED ATTENTION MODULE

During driving in foggy weather, due to the obstructed line of sight, it is difficult to analyze using a single scale. Highly discriminative features can improve the detection performance of the classifier; however, traditional attention modules only focus on spatial and channel dimensions, neglecting the information provided by non-local features. Therefore, multi-scale attention mechanism is applied to driving scenarios. By fusing feature maps of different scales, it is possible to better understand the distribution of obstacles and road conditions around the vehicle. In order to utilize contextual information, this module enhances feature extraction capabilities by introducing self-attention mechanism and multilayer perceptron (MLP), Through clever spatial displacement operations, enabling it to better capture non-local features in the image. This paper proposes an improved S5attention module based on S2-MLP [31], as shown in Figure 6 of this article.

The spatial shift MLP module consists of 4 MLP layers for channel mixing and one mixing patch for spatial shift, as shown in Figure 7.

The input to the spatial-shift layer is a feature X of size $\boldsymbol{x} * \boldsymbol{h} * \boldsymbol{c}$. First, X is evenly divided into four parts along the channel dimension. Then, for each part, a shift operation is performed in four different directions. The formula (5)-(8)

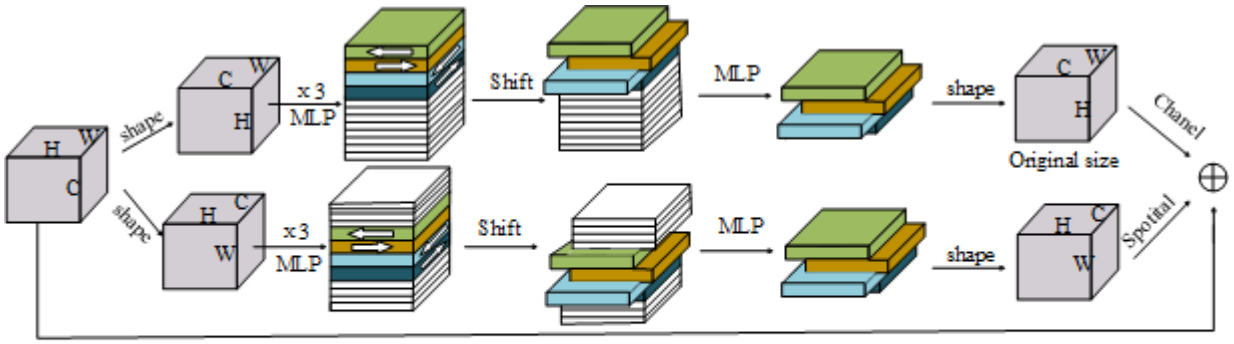

FIGURE 6. S5attention module.

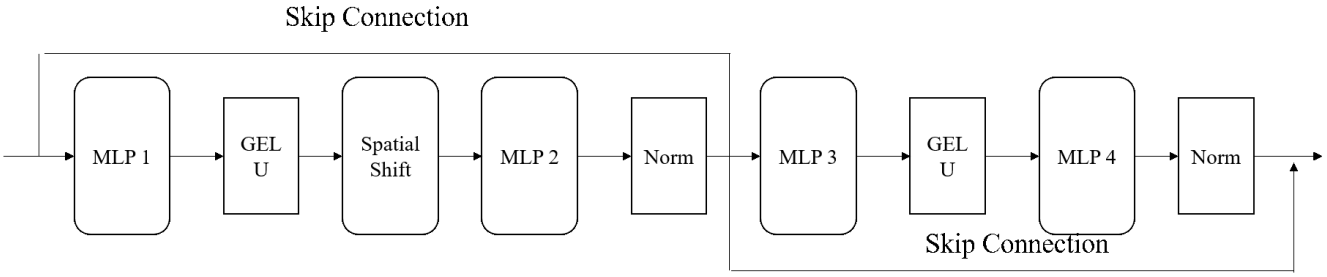

FIGURE 7. Spatial shift MLP module.

represents this process as follows:

$$X\left[2:h,:,1:\frac{c}{4}\right] \leftarrow X\left[1:h-1,:1:\frac{c}{4}\right] \tag{5}$$

$$X\left[1:h-1,:,\frac{c}{4}+1:c/2\right] \leftarrow X\left[2:h,:,1+\frac{c}{4}:\frac{c}{2}\right] \tag{6}$$

$$X\left[:,2:w,\frac{c}{2}:3c/4\right] \leftarrow X\left[:.1:w-1,\frac{c}{2}:3c/4\right] \tag{7}$$

$$X\left[:,1:w-1,\ 3c/4:c\right] \leftarrow X\left[:,2:w,\frac{3c}{4}:c\right] \tag{8}$$

The process is divided into four steps. Firstly, select the first $1/4$ channels $1 : \frac{c}{4}$ of the feature map X in the vertical direction. Move the features in these channels up by one pixel vertically, which means the data from the second row to the h row is replaced by the data from the first row to the h-1 throw. Then, similar operations are performed on the remaining 1/4 channels, respectively, by shifting vertically down, horizontally right, and horizontally left. Through these precise spatial shift operations, we can effectively simulate slight movements of objects in the image, thereby improving the model's robustness to spatial transformations.

III. EXPERIMENTAL RESULTS AND ANALYSIS

A. CONFIGURING THE PARAMETERS AND THE EXPERIMENTAL SETUP

Windows 10 was used as the experimental operating system in this study, and the deep learning models created were frame worked using PyTorch. Table 1 contains particular information on the experimental setup. Stochastic gradient descent (SGD) was used to optimize the training phase. It started with a learning rate of 0.01 and used a cosine annealing hyperparameter of 0.1, a momentum factor of 0.937, and a weight decay coefficient of 0.0005. Training was carried out over 300 epochs with a batch size of 16 and input photos normalized to 640 × 640.

TABLE 1. Experimental environment configuration.

Category	Configuration
CPU	12th Gen Intel(R) Core (TM)i7-12700
Gpu	NVIDIA GeForce RTX 3090 24G
System environment	Windows10
Framework	Pytorch1.13.1
Programming voice	Python3.8

B. EVALUATION INDICATORS

The evaluation indicators chosen for this article include parameters, floating-point operations per second (FLOPs), accuracy, recall, mean average precision (mAP), and frames per second (FPS). The corresponding calculation formulas are as follows:

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

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

$$AP@0.5 = \frac{1}{N}\sum_{i=1}^{n} P_i = \frac{1}{n}P_1 + \frac{1}{n}P_2 + \ldots + \frac{1}{n}P_n \quad (11)$$

$$mAP@0.5:0.95 = \frac{1}{C}\sum_{K=1}^{C} AP@0.5_K \quad (12)$$

$$FPS = \frac{Frames}{Time} \quad (13)$$

Among them, TP (True Positives) represents the bounding boxes correctly detected by the model, FP (False Positives) refers to the bounding boxes falsely detected by the model, and FN (False Negatives) corresponds to the bounding boxes missed by the model.

Precision is a crucial metric. A higher precision value indicates a higher accuracy rate when the model predicts a positive class, meaning the model's prediction results are more credible. Recall, on the other hand, measures the model's ability to identify all positive classes. Especially in traffic driving, a low missed detection rate is particularly critical as it directly relates to driving safety.

Additionally, AP (Average Precision), as a reflection of object detection accuracy, provides us with the model's performance on a single category. When we need to evaluate the comprehensive accuracy of the model for all object recognition, mAP (mean Average Precision) becomes an indispensable metric. It is often used to measure the reliability and overall performance of the model.

Meanwhile, Frames represents the number of frames processed, and Time indicates the detection duration. In autonomous or assisted driving scenarios, real-time detection of road objects requires the model to have a fast response capability to ensure safety and smoothness during driving. Therefore, these metrics together form a comprehensive framework for evaluating the performance of object detection models, providing us with a basis for assessing and improving models from different perspectives.

C. COMPARISON OF DETECTION PERFORMANCE BETWEEN DIFFERENT MODELS

The Faster-RCNN, SSD, YOLOv3, YOLOX, RTMDET [21] and DINO [32] were chosen for comparative trials to assess the effectiveness of the enhanced algorithm suggested in this research. With identical proportions maintained across the training and test sets, these trials were carried out using the same apparatus, dataset, and data augmentation techniques. For testing purposes, the best results from the 200 repetitions of the trials were selected. Table 2 provides comparison information for parameters, recall rate, mAP, precision, and flops.

According to Table 2, YOLOv8s outperforms Faster-RCNN, SSD, YOLOv3, YOLOX, RTMDET and Dino.in terms of accuracy, regression rate, and mAP. Compared to the original YOLOv8s, the improved algorithm YOLO-DF has slightly lower FPS but has a smaller number of parameters and model size, and its accuracy is superior to the original YOLOv8s algorithm.

TABLE 2. Contrast experiment.

Network	P/%	R/%	mAP (%)	Flops/G	Params/M	FPS
Faster-Rcnn	71.6	54.2	58.2	178	41.753	47
SSD	67.8	47.9	53.8	30.63	24.28	60
YOLOv3	44.5	49.1	50.6	15.6	61.95	78
YOLOx	63.4	48.5	63.1	13.323	8.939	87
RTMDET	80.5	79.8	71.6	79.96	52.258	63
Dino	76.5	75.3	67.1	165	40.1	55
YOLOv8s	95.3	89.0	76.9	28.7	11.1	200
Ours	97.5	90.0	81.0	28.1	9.06	166

TABLE 3. Ablation experiment results.

Network	P (%)	R (%)	mAP (%)	Flops	Params
v8s	95.4	89	76.9	28.7	11.1
v8s+c2f-DCN	95.9	88	78.9	25.3	11.4
v8s+S5	95.9	89	79.1	35.2	13.1
v8s+F/I	96.2	89	77.8	21.8	5.9
v8s+ c2f-DCN+S5	98	90	79.5	29.7	13.7
v8s+c2f-DCN+F/I	95.3	90	78.9	20.2	6.3
v8s+S5+F/I	96.2	90	78.4	30.6	8.9
All	97.5	90	81	28.1	9.06

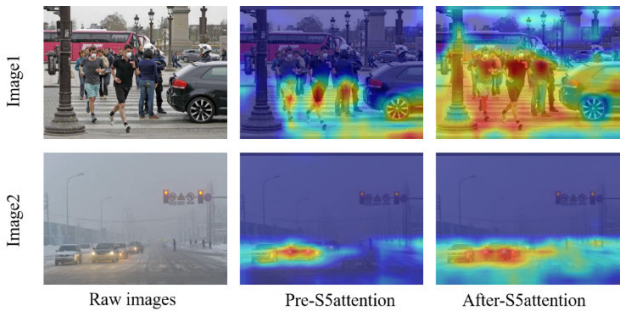

FIGURE 8. Heat map visualizations before and after adding S5attention.

D. ABLATION STUDY

An ablation study was conducted to confirm the efficacy of the proposed improvement methods in this article, and the results are summarized in Table 3. The ablation experiment was divided into 6 groups, with each group maintaining consistency in input images, training hyperparameters, etc. Among them, C2f-DCN, S5attention, FasterNext and Involution(F/I)are the improved methods proposed in this article.

Table 3 reveals that the integration of lightweight models FasterNex and Involution preserves accuracy and recall while significantly reducing the model's weight and improving speed of reasoning, thereby advancing subsequent model deployment. The addition of c2f-DCN increased the model's mAP from 76.9% to 78.9%, further improving accuracy from 95.4% to 97.1%. In contrast, the increase of the S5attention module led to an increase in model parameters and a decrease in network inference speed, but it resulted in a significant increase in the mAP of the algorithm, with the mAP of the test set increasing from 76.9% to 79.1%. Figure 8 provides an intuitive comparison of some detection results before and after adding the S5attention module through heat maps.

Two common scenes from real roads are selected in Figure 8. Image 1 displays a pedestrian crossing, while Image 2 depicts a traffic intersection with traffic lights. Prior to the integration of the S5attention module, observable deficiencies and missed detections were noted in the model's focus areas. Specifically, the model's recognition capability for distant or partially obscured targets required improvement. However, with the incorporation of the designed S5attention module, the model's comprehension of image information has been significantly enhanced. This improvement has enabled the model to better focus on and recognize target information, thereby effectively boosting the overall recognition accuracy and performance.

In this study, a comparative analysis was conducted among seven network models, and it was found that the final improved YOLOv8s showed the most superior overall detection performance. Compared to the original YOLOv8s network, the improved network showed significant enhancements in p-value (2.1%) and mAP (4.1%). The YOLOv8 loss includes classification loss (VFL loss) and regression loss (CIOU loss + distribution focus loss (DFL)), which are weighted by specific weight ratios. The formulas for these losses are as follows:

$$\mathbf{VEL}p, q = \begin{cases} -q(q(\log(\mathbf{p}) + (1-\mathbf{q})\log(1-\mathbf{p}) & \mathbf{q} < 0 \\ -\alpha \mathbf{p}^{\gamma} \log \mathbf{log}(1-\mathbf{p}) & \mathbf{q} = 0 \end{cases} \tag{14}$$

$$\mathcal{L}_{\mathbf{CIoU}} = 1 - \mathbf{IoU} + \frac{\rho^2\left(\mathbf{b}, \mathbf{b}^{\mathbf{gt}}\right)}{\mathbf{c}^2} + \alpha \mathbf{v} \tag{15}$$

$$\mathbf{DFL}\left(\mathbf{S_i}, \mathbf{S_{i+1}}\right) = -\left(\left(\mathbf{y_{i+1}} - \mathbf{y}\right)\log \mathbf{S_i} + \mathbf{log}\left(\mathbf{S_{i+1}}\right)\left(\mathbf{y} - \mathbf{y_i}\right)\right) \tag{16}$$

Among them, the Variant Focal Loss (VFL) is an improved version of Focal Loss, designed to address the issue of class imbalance. When the label q of a sample is less than 0, it means the sample is categorized as a negative sample. In this case, VFL adopts the standard binary cross-entropy loss function to calculate the loss, ensuring that the model does not overfit samples of non-target categories. When q equals 0, it usually indicates that although the sample is labeled as non-target, there may be some uncertainty or ambiguity. A modified form is adopted to reduce the loss of easily classified samples, allowing the model to focus more on those that are difficult to classify or more informative.

IoU represents Intersection over Union, $\boldsymbol{q}$ stands for the label, the center points of the bounding boxes are denoted by $\boldsymbol{b}$ and $\boldsymbol{b}^{gt}$, ρ represents the Euclidean distance between the two bounding boxes, $\boldsymbol{c}$ represents their diagonal distance, $\boldsymbol{v}$ is used to measure the consistency of the relative proportions between the bounding boxes, and α is the weighting coefficient. Firstly, $1 - IoU$ gives the loss component of overlap. The squared Euclidean distance between the predicted and ground truth bounding box centers is calculated and then divided by the squared length of the diagonal of their minimum bounding rectangle, resulting in the loss component for center point deviation, which is adjusted by the weighting coefficient α.

$S_i = \frac{(y_{i+1}-y)}{(y_{y+1}-y)}$, and $S_{i+1} = \frac{(y-y_i)}{(y_{y+1}-y_i)}$ correspond to the prediction probabilities of $\boldsymbol{y_i}$ and $\boldsymbol{y}_{i+1}$. DFL adjusts the probabilities of the two prediction positions $\boldsymbol{y}_{i+1}$ and y_i closest to the true label $\boldsymbol{y}$ in a cross-entropy optimization manner, forcing the network to focus on the positions near the true label $\boldsymbol{y}$. The Train/loss, precision and mAP curves of the seven models are shown in Figure 9.

Figure 9.a show the decreasing trend of the loss function value during the model training process, indicating that the difference between the predicted and actual labels is gradually narrowing, proving the effectiveness of the improved model. Figure 9.b shows that the accuracy of the improved model (ALL) has steadily increased, and compared with other models, the improved model not only achieves higher peak accuracy, but also experiences less fluctuation during the training process. This performance indicates that the improved model encounters less noise interference during the backpropagation process, enabling it to more effectively extract valuable information from the training data, thus achieving higher prediction accuracy. Figure 9.c shows the average precision (mAP) of the models involved in this experiment across different categories, providing a comprehensive evaluation of the overall performance of the model across all target categories. Analyzing the graphical data, it is observed that the improved model (ALL) has a higher mAP value compared to the other seven models, reflecting its higher accuracy and generalization ability.

E. ALGORITHM VERIFICATION

The given text discusses a study that compares seven network models and finds that the improved YOLOv8s has the best overall detection performance. In comparison to the original YOLOv8s, the improved network exhibits significant improvements in p-value and map, and the improved network manifests an enhancement in terms of p-value by 2.1% and an increase in mAP by 4.1%. As shown in Figure 10, the results of actual road target detection by the original model and the improved model are presented.

The comparative experiments in Group A focused on the performance of road object detection in foggy conditions. As evidenced by the experimental results, the original model erroneously identified background as a BUS in the left region, whereas the improved model avoided such misclassifications. Group B's testing scenarios involved the detection of multiple objects with high overlap, where the enhanced algorithm demonstrated superior target localization capabilities, ensuring no missed detections. Furthermore, in Group C's low-visibility tests, even with blurred vehicle images and limited information, the improved model still exhibited remarkable accuracy in recognition. This demonstrates that the enhanced YOLOv8-DF algorithm is capable of handling the challenges of object recognition in complex weather scenarios.

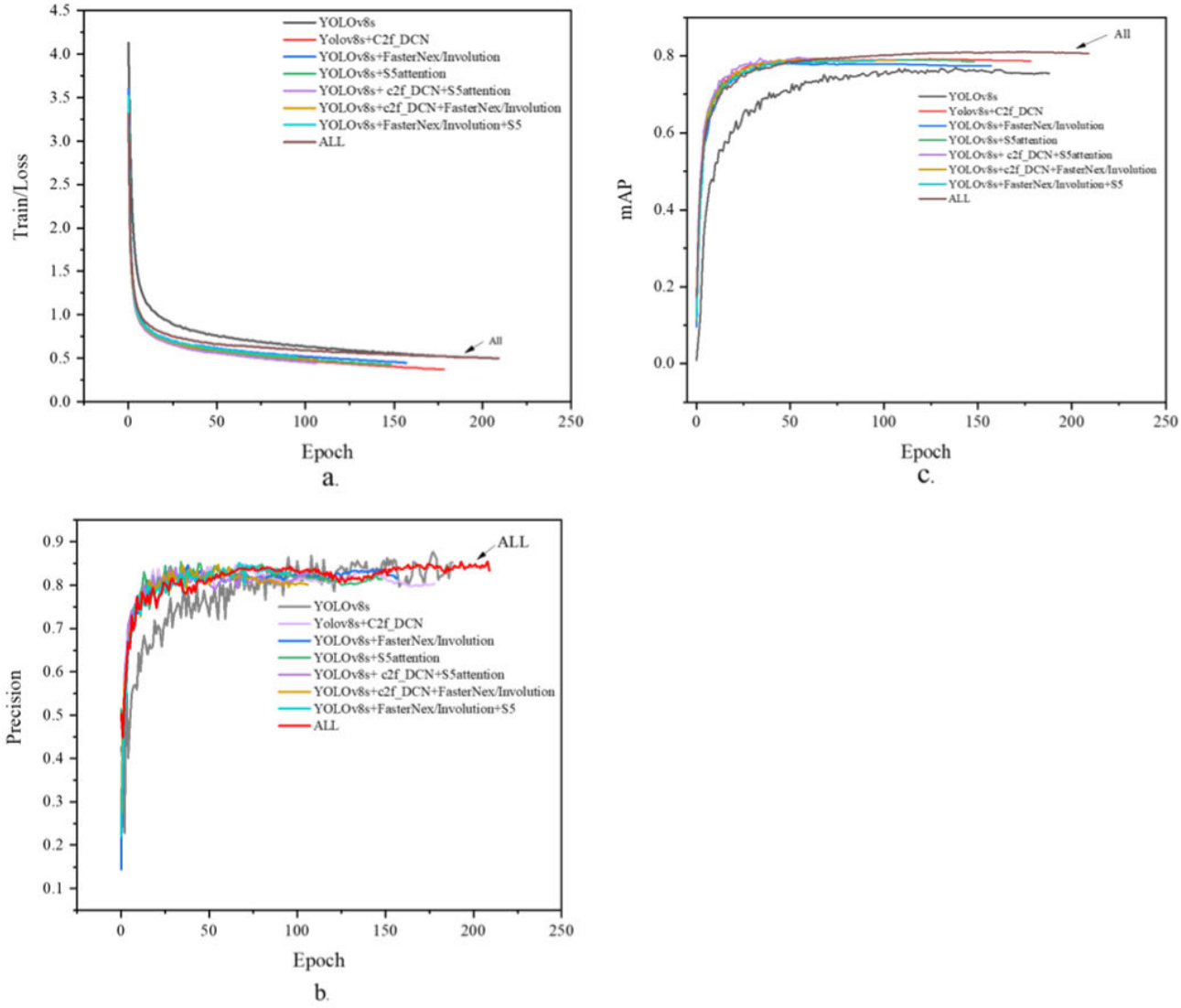

FIGURE 9. a. Training curve, b. Precision curve, c. mAP curve.

FIGURE 10. Comparison of road target detection results.

IV. CONCLUSION

In this paper, an efficient and lightweight YOLOv8-DF network model is proposed for detecting and recognizing traffic targets in foggy weather. By introducing the DCN module and the Involution and FasterNex modules, the model parameters and model size are reduced. A new attention module named S5attention is designed to enhance the feature fusion ability of the model. Additionally, a small target detection layer is added to im-prove the accuracy of detecting small targets, the boundary box regression performance of the network model is improved. Compared with the original network model, the improved YOLOv8-DF network model has a higher accuracy and mAP, with an increase of 2.1% and 4.1% respectively. Moreover, the model parameters and model size are reduced by 0.6 G and 2.04 MB respectively compared to the original network model. Future work will continue to study and improve the network model based on this model, to achieve higher detection accuracy while maintaining fast detection speed. In addition, considering the practical application value of this application direction in real life, edge mobile platform transplantation verification and improvement of the model will be carried out in the future to make the model smaller and easier to deploy.

REFERENCES

[1] C. Li, C. Guo, J. Guo, P. Han, H. Fu, and R. Cong, "PDR-Net: Perception-inspired single image dehazing network with refinement," *IEEE Trans. Multimedia*, vol. 22, no. 3, pp. 704–716, Mar. 2020, doi: 10.1109/TMM.2019.2933334.
[2] X. Xiaomin and L. Wei, "Two stages end-to-end generative network for single image defogging," *J. Comput.-Aided Des. Comput. Graph.*, vol. 32, no. 1, pp. 164–172, 2020, doi: 10.3724/SP.J.1089.2020.17856.
[3] J. Wu, Z. Kuang, L. Wang, W. Zhang, and G. Wu, "Context-aware RCNN: A baseline for action detection in videos," 2020, *arXiv:2007.09861*.
[4] L. Jiang, J. Chen, H. Todo, Z. Tang, S. Liu, and Y. Li, "Application of a fast RCNN based on upper and lower layers in face recognition," *Comput. Intell. Neurosci.*, vol. 2021, pp. 1–12, Sep. 2021.
[5] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 1440–1448.
[6] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," 2017, *arXiv:1703.06870*.
[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2015, *arXiv:1506.02640*.
[8] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," 2016, *arXiv:1612.08242*.
[9] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
[10] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.
[11] M. Wang, W. Yang, L. Wang, D. Chen, F. Wei, H. KeZiErBieKe, and Y. Liao, "FE-YOLOv5: Feature enhancement network based on YOLOv5 for small object detection," *J. Vis. Commun. Image Represent.*, vol. 90, Feb. 2023, Art. no. 103752, doi: 10.1016/j.jvcir.2023.103752.
[12] C.-Y. Wang, A. Bochkovskiy, and H.-Y.-M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Vancouver, BC, Canada, Jun. 2023, pp. 7464–7475.
[13] L. Wei, A. Dragomir, E. Dumitru, S. Christian, R. Scott, F. Cheng-Yang, B. Alexander, "SSD: Single shot MultiBox detector," in *Computer Vision—ECCV 2016*, vol. 9905. Cham, Switzerland: Springer, 2016.
[14] B. Gao, Z. Jiang, and J. Zhang, "Traffic sign detection based on SSD," in *Proc. 4th Int. Conf. Autom., Control Robot. Eng.*, Jul. 2019, p. 16, doi: 10.1145/3351917.3351988.
[15] L. Xuan, L. Jing, and W. Haiyan, "Study on traffice scene object detection algorithm under complex meteorological conditions," *Comput. Simul.*, vol. 38, no. 2, pp. 87–90, 2021.
[16] W. Qi-Ming, Z. He, D. Zhang, and Z. Mao, "Research on pedestrian and vehicle detection method based on YOLOv3 in foggy scene," *Control Eng. China*, vol. 1, pp. 1–8, Sep. 2023.
[17] J. Zetao, X. Yun, and Z. Shaoqin, "Low-illumination object detection method based on dark-YOLO," *J. Comput.-Aided Des. Comput. Graph.*, vol. 35, no. 3, pp. 441–451, 2023.
[18] J. Yin, S. Qu, Z. Yao, X. Hu, X. Qin, and P. Hua, "Traffic sign recognition model in haze weather based on YOLOv5," *J. Comput. Appl.*, vol. 42, no. 9, pp. 2876–2884, 2022.
[19] K. Li, Y. Wang, and Z. Hu, "Improved YOLOv7 for small object detection algorithm based on attention and dynamic convolution," *Appl. Sci.*, vol. 13, no. 16, p. 9316, Aug. 2023, doi: 10.3390/app13169316.
[20] C. Bhagya and A. Shyna, "An overview of deep learning based object detection techniques," in *Proc. 1st Int. Conf. Innov. Inf. Commun. Technol. (ICIICT)*, Chennai, India, Apr. 2019, pp. 1–6.
[21] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable transformers for end-to-end object detection," 2020, *arXiv:2010.04159*.
[22] B. Li, W. Ren, D. Fu, D. Tao, D. Feng, W. Zeng, and Z. Wang, "Benchmarking single-image dehazing and beyond," *IEEE Trans. Image Process.*, vol. 28, no. 1, pp. 492–505, Jan. 2019.
[23] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.
[24] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*.
[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. NIPS*, 2017, pp. 1–11.
[26] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," 2017, *arXiv:1703.06211*.
[27] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable ConvNets v2: More deformable, better results," 2018, *arXiv:1811.11168*.
[28] Y. Guo, Y. Li, R. Feris, L. Wang, and T. Rosing, "Depthwise convolution is all you need for learning multiple visual domains," 2019, *arXiv:1902.00927*.
[29] J. Chen, S.-H. Kao, H. He, W. Zhuo, S. Wen, C.-H. Lee, and S.-H.-G. Chan, "Run, don't walk: Chasing higher FLOPS for faster neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 12021–12031.
[30] S. Park, Y.-J. Yeo, and Y.-G. Shin, "PConv: Simple yet effective convolutional layer for generative adversarial network," *Neural Comput. Appl.*, vol. 34, no. 9, pp. 7113–7124, May 2022, doi: 10.1007/s00521-021-06846-2.
[31] T. Yu, X. Li, Y. Cai, M. Sun, and P. Li, "S2-MLP: Spatial-shift MLP architecture for vision," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)* Jan. 2022, pp. 3615–3624.
[32] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. M. Ni, and H.-Y. Shum, "DINO: DETR with improved de noising anchor boxes for end-to-end object detection," 2022, *arXiv:2203.03605*.

LONG CHENG is currently pursuing the master's degree with the School of Automotive and Traffic Engineering, Jiangsu University of Technology, Changzhou, China. His current research interests include multi-target tracking and recognition path planning.

DAN ZHANG received the Ph.D. degree in mechanical engineering from Yamagata University, Yamagata, Japan, in 2015. Since 2015, she has been an Assistant Professor with the Vehicle Engineering Department, Jiangsu University of Technology. Her current research interests include pattern recognition and auto-autonomous driving.

YAN ZHENG received the Ph.D. degree in mechanical engineering from Yamagata University, Yamagata, Japan, in 2014. He is currently a Professor with the Vehicle Engineering Department, Jiangsu University of Technology. His current research interests include machine learning and its application in vehicle engineering.