EAFYOLO: A Efficient Yolo For Autonomous Object Detection in Foggy Weather

XinZhao Gao

***Abstract*—Due to the significant reduction in visibility during foggy weather, it poses a considerable challenge for autonomous driving object detection in foggy conditions. There are numerous existing methods to enhance detection performance; however, they do not simultaneously address the issues of detection accuracy, detection speed, and the scarcity of labeled datasets.This paper proposes a rapid detection algorithm based on YOLOv10 and unsupervised domain adaptation for autonomous driving object detection in foggy weather. Firstly, to tackle the issue of limited labeled datasets, the concept of unsupervised domain adaptation is employed, utilizing adversarial learning to achieve domain alignment between foggy and clear weather conditions. This allows for training with only labeled clear weather images and unlabeled foggy images. Secondly, as most current achievements focus solely on detection performance, the paper addresses the speed issue by adopting YOLOv10 without non-maximum suppression (NMS) post-processing as the fundamental detection framework, significantly improving inference speed. Lastly, due to the anchor-free and NMS-free nature of the network, the generalization performance may decline, especially for small-sized targets. The paper replaces the CIOU-based box loss with WNWDLoss to enable the network to handle targets of various sizes, thereby enhancing detection accuracy. Experimental results on the public datasets Cityscapes, FoggyCityscapes and KITTI demonstrate the effectiveness of the proposed method.**

***Index Terms*—Autonomous driving object detection, Unsupervised domain adaptation, Domain alignment, Foggy weather conditions**

I. INTRODUCTION

Driven by deep learning, the performance of object detection has been continuously increasing and has achieved remarkable results in the field of autonomous driving. However, under adverse weather conditions such as fog, the performance of detection algorithms for autonomous vehicles significantly declines due to low visibility, potentially leading to traffic hazards. In fact, this drastic performance drop can be attributed to the domain shift between foggy and clear weather images. Images of clear weather with manual annotations serve as the source domain for supervised training to obtain object detection models. However, models trained on the source domain fail to extract correct features from foggy images, which are the target domain. Inspired by[1], we utilize adversarial learning to achieve feature alignment by incorporating pixel-level domain adaptation modules and instance-level domain adaptation modules. Most current work is based on region proposal-based two-stage detectors, which, although improved, are much slower than one-stage detectors. Therefore, we introduce the feature alignment module into the one-stage detector, YOLOv10 [2]. Additionally, we find that image defogging and enhancement contribute more to the accuracy of the detector than the stacking of detector modules. Based on [3], we propose an O(1)-level image defogging prior knowledge module with no additional trainable parameters, which, when used in the image preprocessing stage during network inference, can significantly increase detection performance. Due to occlusions, small targets at long distances in autonomous driving are more likely to be obscured. To enhance the detection performance of small-sized targets, we replace the box loss with a combination of CIoU loss and NWD loss, referred to as WNWD loss, to balance the attention to targets of different sizes at different distances in foggy conditions during autonomous driving.

In summary, the contributions of this paper are as follows:

We propose a one-stage foggy weather autonomous driving rapid detection model based on domain adaptation, incorporating pixel-level and instance-level domain adaptation modules for detectors of three sizes. Training with labeled clear weather data and unlabeled foggy weather data to enhance the performance and robustness of foggy weather object detection.

We introduce an O(1)-level image defogging prior module with no additional trainable parameters, which, when used as an image preprocessing operation during detection can significantly improves the network's performance.

To address the occlusion issue of small targets at medium and long distances, we combine CIoU loss with NWD loss to form WNWD loss, balancing the focus on targets of different sizes at different distances in foggy conditions during autonomous driving.

All our improvements incur no additional trainable parameters during inference, maintaining the same network parameter count and inference time as YOLOv10, surpassing other detectors in the field with the same detection accuracy.

II. METHOD

In this paper, for the sake of convenience, we denote N labeled clear weather images (source domain) as $S = \{X_s^i, L_s^i\}_{i=1}^N$, and N unlabeled foggy weather images (target domain) as $T = \{X_t^i\}_{i=1}^N$, where X_s^i, L_s^i, and X_t^i represent the i-th image of the source domain, the annotation information corresponding to the i-th image of the source domain, and the i-th image of the target domain, respectively.

Manuscript received March 21, 2025

XinZhao Gao, School of computer science and technology, Tiangong University, Tianjin, China

A. Network Architecture

Our object detector is based on the NMS-free YOLOv10, with the overall architecture as shown in Fig. 1. During training, images from the target domain first pass through the ImagePreDefogModel (IPDM) defogging prior module and are then combined with source domain images that are processed by the feature extractor to generate features at three different scales. These features simultaneously enter the neck section and our pixel-level domain adaptation module (PDA) of this paper. Subsequently, the three sizes of features after feature fusion by the neck section enter the detection head and our object-level domain adaptation module (ODA) of this paper, producing prediction results.

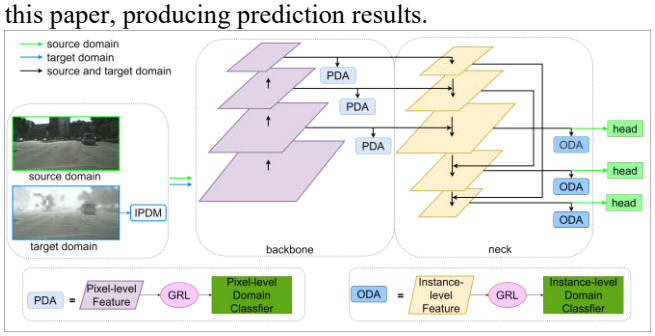

Fig. 1 Network architecture

The Gradient Reversal Layer (GRL) [4] is located between the feature extractor and the domain classifier. It maintains the gradient unchanged during the normal forward propagation of the network, and during backpropagation, it reverses the gradient of the domain classifier's loss, changing the direction of parameter updates. This makes it more difficult for the domain classifier to distinguish the domain category of the input features, thereby extracting domain-invariant features and achieving the purpose of domain adaptation.

During inference, the network pipeline is the same as the original network, with the addition of the cost-free IPDM in the preprocessing stage. It can be seen that our network has essentially the same number of parameters and running time as the original network during inference. To better understand the process by which the domain adaptation module acts on the network through GRL, we detail the gradient flow and loss calculation process of the network through Fig. 2.

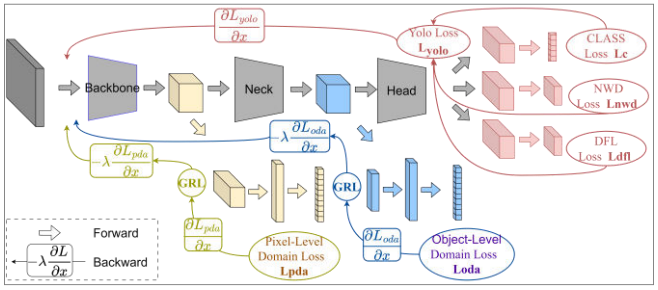

Fig. 2 Forward propagation and backpropagation

B. Domain Adaption Model

The domain adaptation module in this paper includes a Pixel-Level Domain Adaptation (PDA) module and an Object-Level Domain Adaptation (ODA) module. The Pixel-Level Domain Adaptation module is located after the backbone, which contains rich global information, and classifies through a pixel-level domain classifier composed of several convolutional layers and activation layers, which then feeds back into the network to achieve the extraction of domain-invariant features. Our PDA loss is shown in Equation (1).

$$L_{PDA} = -(1/NWH)\sum_{i=1}^{N}\sum_{j=1}^{W\times H} y_i \log(G_{PDA}(x_i)_j) + (1-y_i)\log(1-G_{PDA}(x_i)_j) \quad (1)$$

In the given context, $x_i \epsilon \{x_1, x_2, x_3, \dots, x_N\}$ represents the N input images, W and H are the width and height of the i-th image, $y_i \epsilon \{0,1\}$, indicating the domain label of the i-th image, where 0 represents the source domain and 1 represents the target domain. G_{PDA} is the domain classifier of the PDA module, and $G_{PDA}(x_i)_j$ represents the j-th pixel value in the domain classifier result for the i-th image.

In the given context, $x_i \epsilon \{x_1, x_2, x_3, \dots, x_N\}$ represents the N input images, W and H are the width and height of the i-th image, $y_i \epsilon \{0,1\}$, indicating the domain label of the i-th image, where 0 represents the source domain and 1 represents the target domain. G_{PDA} is the domain classifier of the PDA module, and $G_{PDA}(x_i)_j$ represents the j-th pixel value in the domain classifier result for the i-th image.

Since the information extracted by the network backbone contains relatively fewer local features, deep and shallow information are fused together in the feature fusion part of the network. Therefore, we add the Object-Level Domain Adaptation (ODA) module after the feature fusion module that contains rich instance information and perform binary classification through some linear layers. Our ODA loss is shown in Equation (2):

$$L_{ODA} = (-1/N)\sum_{i=1}^{N}\begin{pmatrix} y_i \log(G_{ODA}(x_i)) + \\ (1-y_i)\log(1-G_{ODA}(x_i)) \end{pmatrix} \quad (2)$$

Due to the network backbone having three different scales of feature outputs, correspondingly, we add a PDA module for each scale of output. Similarly, the neck section fuses features of three different scales before they enter the detection heads corresponding to their respective sizes. Our three ODA modules are also placed in front of the detection heads for each size to perform feature alignment on the feature fusion part. Therefore, our domain adaptation part loss is defined as Equation (3), where λ_{PDA} and λ_{ODA} are hyperparameters. We set $\lambda_{PDA} = 1.0$ and $\lambda_{ODA} = 2.0$.

$$L_{UDA} = \lambda_{PDA}\sum_{i=1}^{3} L_{PDAi} + \lambda_{ODA}\sum_{i=1}^{3} L_{ODAi} \quad (3)$$

C. IPDM

Currently, UDA-based foggy weather object detection methods are dedicated to the improvement of the network itself, striving to better learn the invariant information between the source and target domains. However, we found that after a certain point, the enhancement of the network's feature extraction performance obtained by stacking ordinary modules and increasing the number of parameters is less than the improvement in network performance brought about by strong image enhancement. Therefore, starting from the problems of the task itself, we propose a simple prior module IPDM based on traditional image defogging algorithms. Our goal is to increase network performance with negligible

inference speed without having a target domain dataset, hence deep learning-based defogging algorithms are not considered. We base our approach on the lightweight but effective CLAHE (Contrast Limited Adaptive Histogram Equalization)[3] and propose a defogging prior module IPDM with almost no runtime.

To validate the advantages of our IPDM, subsequent experimental sections will provide a comparison of the performance and runtime of different defogging modules on the network.

D. WNWD Loss

In foggy weather autonomous driving object detection, due to occlusions, objects that are farther away appear smaller and are more severely occluded. This is very unfavorable for the vast majority of detectors that base their judgment of the distance and similarity between two boxes on the Intersection over Union (IOU) metric, as IOU is highly sensitive to the errors of small objects. Additionally, due to YOLOv10's NMS-free nature, although the network's inference speed has been greatly improved, it also increases instability across different tasks, duplicate detections, missed detections, and false positives.

Therefore, in order to enhance the network's focus on distant small targets and dense targets, we replace the original box loss with the Normalized Gaussian Wasserstein Distance (NWD) [5]. This increases the detection performance for small targets at a distance. The core idea of NWD is to model the bounding box as a two-dimensional Gaussian distribution, specifically represented as Equation (4), with weights that decrease from the center of the bounding box outward.

$$\mu = \begin{bmatrix} c_x \\ c_y \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \frac{w^2}{4} & 0 \\ 0 & \frac{h^2}{4} \end{bmatrix} \tag{4}$$

In which c_x, c_y, w, h represents the center coordinates and the width and height of the box, respectively. Ultimately, the NWD between two boxes B_1 and B_2 can be expressed as Equation (5), where $\|\cdot\|_F$ denotes the Frobenius norm.

$$W_2^2(B_1, B_2) = \| \mu_1 - \mu_2 \|_2^2 + \| \Sigma_1^{1/2} - \Sigma_2^{1/2} \|_F^2 \tag{5}$$

However, we found that the NWD loss performs worse on medium to large targets compared to the original network's CIoU loss. Therefore, we made a weighted sum of these two parts of the loss, thereby enabling the detector to balance the detection of targets of various sizes.

Finally, the overall loss function of the detector is defined as Equation (6), where the bolded parts represent the loss components proposed or modified in this paper, and the other three losses are the inherent loss functions of YOLOv10.

$$\begin{aligned} L = {} & \boldsymbol{\lambda_1} \cdot L_{cls} + (\boldsymbol{\lambda_{nwd}} \cdot \mathbf{L_{nwd}} + \boldsymbol{\lambda_{ciou}} \cdot L_{ciou}) \\ & + \boldsymbol{\lambda_2} \cdot L_{dfl} + \boldsymbol{\lambda_3} \cdot \mathbf{L_{UDA}} \end{aligned} \tag{6}$$

III. Experiment

A. Dataset Preparation

Cityscapes[6]: The Cityscapes dataset is an autonomous driving semantic segmentation dataset focusing on urban street scenes. It comprises images recorded from street scenes in 50 different cities, with each image annotated at the pixel level to identify various objects and scene categories. It includes 2,975 finely annotated training images and 500 finely annotated validation images. Cityscapes fully reflects the complexity of real clear-weather urban traffic scenarios and serves as our source domain dataset.

Foggy Cityscapes[7]: By denoising and completing the computed raw depth maps and using an optical scattering model for fog synthesis, which is then added to the images in the Cityscapes dataset, the Foggy Cityscapes dataset was established to simulate fog in real scenes. Each foggy image is rendered using clear images and depth maps from Cityscapes. The annotations and data split in Foggy Cityscapes are inherited from Cityscapes, and for each original image, three different fog density levels with coefficients of 0.005, 0.01, and 0.02 are generated. We select the images with the lowest visibility, i.e., the coefficient of 0.02, as the target domain.

KITTI[8]: The KITTI dataset, jointly created by the Karlsruhe Institute of Technology and the Toyota Technical Institute in the USA, is currently the largest computer vision benchmark dataset for autonomous driving scenarios. Its data acquisition platform is equipped with 2 grayscale cameras, 2 color cameras, a Velodyne 64-line 3D LiDAR, 4 optical lenses, and a GPS navigation system. We utilize its 2D object detection information, which includes 7,481 training images, 7,518 test images, and corresponding point cloud data, totaling 80,256 annotated objects.

B. Training Procedure and Hyperparameter Settings

The model is based on YOLOv10m, and during training, we simultaneously use both the source and target domain datasets. The source domain dataset participates in normal detection training and forward propagation, with a domain label of 0 added to the domain classifier to calculate the domain classification loss, which is propagated back along with the detector's loss. The target domain dataset only has a domain label of 1 added to the domain classifier to calculate the domain classification loss, without passing through the detector's heads, nor participating in the detector's loss calculation or backpropagation.

We use the YOLOv10m network with the same parameter settings as EAFyolo as the baseline network, trained only on the source domain and then tested on the target domain, with the results serving as the baseline. We also conduct supervised training of this network on the target domain, with the results serving as the oracle.

In the experiments, for image preprocessing, the input images are resized to the default dimension of 640*640, with mosaic set to 1.0, mixup to 0.05, and crop_fraction to 1.0. We optimize the network using the AdamW optimizer that comes with YOLOv10 and train for 200 epochs, with lrf set to 0.1, momentum to 0.9, and weight_decay to 0.0005. The corresponding learning rate lr=0.000833 is automatically calculated based on the expected number of detected categories using its own lr formula. For the hyperparameters of the modules proposed in this paper, we set λ_PDA=1.0, λ_ODA=2.0, λ_CIOU=0.9, λ_NWD=1.9.

All training and inference experiments are conducted on a single NVIDIA GeForce RTX 4090 (24GB memory).

C. Cityscapes to Foggy Cityscapes Results

The detection results of EAFyolo from Cityscapes to Foggy Cityscapes are shown in Table Ⅰ. It can be observed that our network achieved better results, with a mAP of 50.9%, surpassing the performance of other detectors. The performance improvement over the baseline is 5.9% in AP, and it is only 2.5% AP behind the oracle based on supervised learning, significantly enhancing the detector's performance when data is scarce. RYOLO used QTNet for dataset augmentation by converting between normal and foggy conditions, whereas our detector achieved a higher mAP without additional data augmentation. However, both of these detectors are based on the two-stage Faster R-CNN model and rely on pseudo-labeling to enhance detection performance, which increases network complexity and affects the real-time performance of object detection.

Table I Detection results from Cityscapes to Foggy Cityscapes

Method	Map
ParaUDA [9]	41.7
DADetect [10]	42.3
PT [11]	42.7
IDF [12]	42.4
SIGMA [13]	44.2
RYOLO [14]	49.5
CMT [15]	50.3
MILA [16]	50.6
baseline [2]	45.0
ours	50.9
oracle	53.4

D. Cross-CARMA Adaptation Results

To validate the effectiveness of the detector and to avoid the contingency of the modules within this detector, we employ our detector for cross-domain adaptation detection between autonomous driving datasets captured by different cameras. Here, we select the real-world captured 3D dataset KITTI as the source domain, focusing on the category with the most instances, cars, for domain adaptation. We conduct domain adaptation from KITTI to Cityscapes. The results, as shown in Table Ⅱ, demonstrate that our detector outperforms other detectors with a performance of 60.3% AP and shows an improvement of 7.5% AP over a regular detector that does not utilize the modules of this detector.

Table II Detection results from KITTI to Cityscapes

Method	AP
IDF [12]	42.1
MeGACDA [17]	43.0
TIA [18]	44.0
ParaUDA [9]	43.6
EPM [19]	45.0
SIGMA [13]	45.8
base	52.8
ours	60.3

E. Module Ablation Study

We conducted ablation experiments to validate the performance of each module, using a YOLOv10 network with the same parameters as EAFyolo as the base model. We incrementally added the modules proposed in this paper to compare their performance. To better contrast the impact of each module on the detector, we combined the three modules in all possible ways, resulting in a total of 8 combinations. The experimental results are shown in Table Ⅲ.

Compared to the base model without any modules, IPDM can improve the AP by 1.2%, WNWDloss can enhance the AP by 1.5%, and UDA can boost the AP by 3.8%. It is evident that among these three modules, UDA contributes the most significant enhancement to detection performance. Furthermore, among the combinations of two modules, IPDM + WNWDloss can improve the AP by 3.2%, IPDM + UDA can enhance the AP by 4.5%, and UDA + WNWDloss can increase the AP by 4.6%. Thus, it is observable that the performance improvement from any two-module combination is greater than that of a single module. Lastly, when all three modules are utilized, the AP can be improved by 5.9%, which is higher than the performance improvement of any single module or two-module combination.

Table III The ablation experiment results of each module from Cityscapes to FoggyCityscapes

Model	IPDM	UDA	WNWD loss	Map
Base				45.0
Base +IPDM	√			46.2
Base +WNWDloss			√	46.5
Base +IPDM+ WNWDloss	√		√	48.2
Base +UDA		√		48.8
Base +IPDM+UDA	√	√		49.5
Base +UDA+ WNWDloss		√	√	49.6
Base +IPDM+UDA+ WNWDloss	√	√	√	50.9

F. Different Defogging Methods for IPDM

In the network image enhancement defogging module section, let us first recall that our goal is to achieve the highest detection performance with as few additional parameters as possible. Therefore, we have chosen a traditional image enhancement-based defogging module. Although its performance may be lower compared to current deep learning-based defogging methods, its inference speed is almost negligible compared to the network, and its combination with subsequent UDA and WNWD modules significantly enhances the network's detection performance.

Table IV The performance of different image enhancement modules on our network

Defog method	Map
Single Scale Retinex [20]	41.7
Adaptive Histogram Equalization [21]	46.5
Multi-Scale Retinex with Color Restoration [22]	46.5
Adaptive Gamma Correction [23]	49.6
Multi-Scale Retinex [24]	49.7
Dark Channel Prior [25]	50.7
Contrast Limited Adaptive Histogram Equalization (used)	50.9

Here, we selected seven classic image enhancement defogging algorithms to test as the core content of IPDM. The detection results are shown in Table Ⅳ. The results indicate that the methods with better performance are Dark Channel Prior[10] and CLAHE, both with accuracy rates above 50% AP, differing by only 0.2% AP. The defogging effects can also be clearly seen in Fig. 3. However, the Dark Channel Prior method can produce some color shifts, which may subsequently affect the detection. Therefore, we chose CLAHE, which has fast execution speed and good performance, as the core method for the image enhancement defogging module.

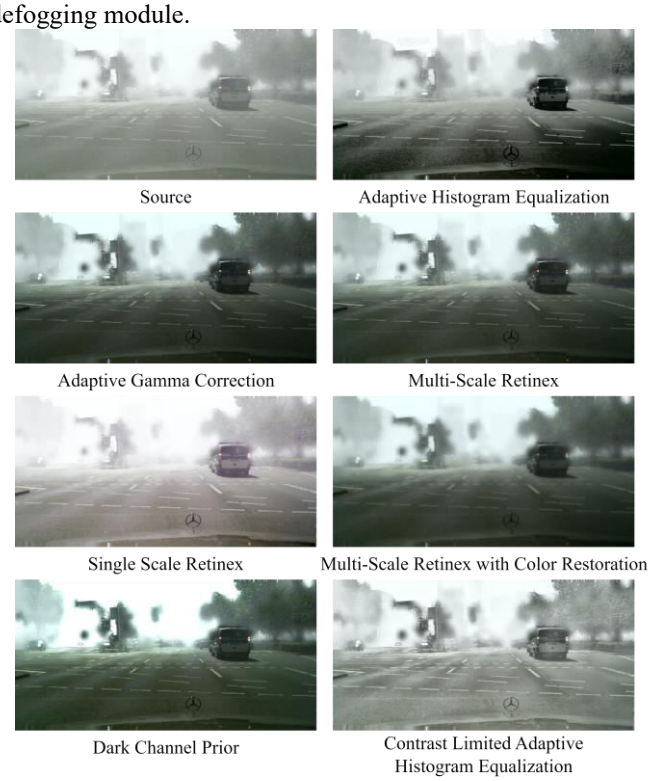

Fig. 3 Visual comparison of dehazing effects of different image enhancement modules on the Foggy Cityscapes dataset

IV. CONCLUSION

This paper addresses the issue of limited data quantity and low performance in foggy weather autonomous driving object detection by proposing a novel detection framework, EAFyolo. It employs unsupervised domain adaptation to tackle the problem of scarce training data. Additionally, it integrates traditional fast image enhancement defogging algorithms, enabling the detector to better learn knowledge in foggy conditions without additional parameter overhead or increased runtime. To address the issue of low visibility and poor detection performance of small targets in foggy weather, WNWDloss is utilized to balance the detection of objects of various sizes, thereby enhancing the network's robustness. We validate the effectiveness of each module and the overall performance of the detector through experiments from Cityscapes to Foggy Cityscapes, KITTI to Cityscapes, and ablation studies of individual modules. The experimental data demonstrate the efficacy of these methods.

REFERENCES

[1] Y. Ganin et al., “Domain-adversarial training of neural networks,” J.Mach. Learn. Res., vol. 17, no. 1, pp. 2096–2030, 2016.
[2] Wang A, Chen H, Liu L, et al. Yolov10: Real-time end-to-end object detection[J]. arxiv preprint arxiv:2405.14458, 2024.
[3] Zuiderveld K. Contrast limited adaptive histogram equalization[M]//Graphics gems IV. 1994: 474-485.
[4] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In International Conference on Machine Learning, pages 1180–1189. PMLR, 2015.
[5] Wang J, Xu C, Yang W, et al. A normalized Gaussian Wasserstein distance for tiny object detection[J]. arxiv preprint arxiv:2110.13389, 2021.
[6] Cordts M, Omran M, Ramos S, et al. The cityscapes dataset for semantic urban scene understanding[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 3213-3223.
[7] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. International Journal of Computer Vision, 126:973–992, 2018.
[8] A. Geiger, P. Lenz and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 2012, pp. 3354-3361, doi: 10.1109/CVPR.2012.6248074.
[9] Zhang W, Wang J, Wang Y, et al. Parauda: Invariant feature learning with auxiliary synthetic samples for unsupervised domain adaptation[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(11): 20217-20229.
[10] Li J, Xu R, Ma J, et al. Domain adaptive object detection for autonomous driving under foggy weather[C]//Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2023: 612-622.
[11] Chen M, Chen W, Yang S, et al. Learning domain adaptive object detection with probabilistic teacher[J]. arxiv preprint arxiv:2206.06293, 2022.
[12] Lang Q, Zhang L, Shi W, et al. Exploring implicit domain-invariant features for domain adaptive object detection[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2022, 33(4): 1816-1826.
[13] Li W, Liu X, Yuan Y. Sigma: Semantic-complete graph matching for domain adaptive object detection[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 5291-5300.
[14] Wang L, Qin H, Zhou X, et al. R-YOLO: A robust object detector in adverse weather[J]. IEEE Transactions on Instrumentation and Measurement, 2022, 72: 1-11.
[15] Cao S, Joshi D, Gui L Y, et al. Contrastive mean teacher for domain adaptive object detectors[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023: 23839-23848.
[16] Krishna O, Ohashi H, Sinha S. MILA: memory-based instance-level adaptation for cross-domain object detection[J]. arxiv preprint arxiv:2309.01086, 2023.
[17] Vs V, Gupta V, Oza P, et al. Mega-cda: Memory guided attention for category-aware unsupervised domain adaptive object detection[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 4516-4526.
[18] Zhao L, Wang L. Task-specific inconsistency alignment for domain adaptive object detection[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022: 14217-14226.
[19] Hsu C C, Tsai Y H, Lin Y Y, et al. Every pixel matters: Center-aware feature alignment for domain adaptive object detector[C]//Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16. Springer International Publishing, 2020: 733-748.
[20] Choi D H, Jang I H, Kim M H, et al. Color image enhancement using single-scale retinex based on an improved image formation model[C]//2008 16th European Signal Processing Conference. IEEE, 2008: 1-5.
[21] Pizer S M, Amburn E P, Austin J D, et al. Adaptive histogram equalization and its variations[J]. Computer vision, graphics, and image processing, 1987, 39(3): 355-368.
[22] Parthasarathy S, Sankaran P. An automated multi scale retinex with color restoration for image enhancement[C]//2012 National Conference on Communications (NCC). IEEE, 2012: 1-5.
[23] Rahman S, Rahman M M, Abdullah-Al-Wadud M, et al. An adaptive gamma correction for image enhancement[J]. EURASIP Journal on Image and Video Processing, 2016, 2016: 1-13.

[24] Rahman Z, Jobson D J, Woodell G A. Multi-scale retinex for color image enhancement[C]//Proceedings of 3rd IEEE international conference on image processing. IEEE, 1996, 3: 1003-1006.

[25] HE K · SUN J · TANG X J · et al.Single image haze removal using dark channel prior[J].IEEE Transactions on Pattern Analysis and Machine Intelligence · 2010 · 33（12）：2341-2353.