mathematics

MDPI

Article

Joint Semantic Deep Learning Algorithm for Object Detection under Foggy Road Conditions

Mingdi Hu [1,*], Yixuan Li [1], Jiulun Fan [1] and Bingyi Jing [2,*]

[1] School of Communications and Information Engineering, Xi'an University of Posts and Telecommunications, Chang'an West St., Xi'an 710121, China

[2] Department of Statistics & Data Science, Southern University of Science and Technology, 1088 Xueyuan Avenue, Shenzhen 518055, China

* Correspondence: humingdi2005@xupt.edu.cn (M.H.); jingby@sustech.edu.cn (B.J.)

Abstract: Current mainstream deep learning methods for object detection are generally trained on high-quality datasets, which might have inferior performances under bad weather conditions. In the paper, a joint semantic deep learning algorithm is proposed to address object detection under foggy road conditions, which is constructed by embedding three attention modules and a 4-layer *UNet* multi-scale decoding module in the feature extraction module of the backbone network *Faster RCNN*. The algorithm differs from other object detection methods in that it is designed to solve low- and high-level joint tasks, including dehazing and object detection through end-to-end training. Furthermore, the location of the fog is learned by these attention modules to assist image recovery, the image quality is recovered by *UNet* decoding module for dehazing, and then the feature representations of the original image and the recovered image are fused and fed into the *FPN* (Feature Pyramid Network) module to achieve joint semantic learning. The joint semantic features are leveraged to push the subsequent network modules ability, and therefore make the proposed algorithm work better for the object detection task under foggy conditions in the real world. Moreover, this method and *Faster RCNN* have the same testing time due to the weight sharing in the feature extraction module. Extensive experiments confirm that the average accuracy of our algorithm outperforms the typical object detection algorithms and the state-of-the-art joint low- and high-level tasks algorithms for the object detection of seven kinds of objects on road traffics under normal weather or foggy conditions.

Keywords: machine learning; deep convolutional neural network; object detection; joint semantic deep learning; single image fog removal

MSC: 54H30; 68U10; 94A08

check for updates

Citation: Hu, M.; Li, Y.; Fan, J.; Jing, B. Joint Semantic Deep Learning Algorithm for Object Detection under Foggy Road Conditions. *Mathematics* **2022**, *10*, 4526. https://doi.org/10.3390/math10234526

Academic Editor: Andrey Gorshenin

Received: 27 October 2022
Accepted: 25 November 2022
Published: 30 November 2022

1. Introduction

Fog is a common weather occurrence and can severely damage the image quality captured by the outdoor equipment. There has been a large body of literature on object detection under inclement weather conditions, which include one-stage and combination approaches [1,2]. For example, one-stage approaches include the domain-based adaptive target detection algorithm under foggy conditions [3–8]; however, these methods also have limitations, for example, their performance is not guaranteed when the training and test data sets are vastly different, and these methods failed to take advantage of the image recovery potential information while combination ones include single image fog removal [9–12] and combined object detection [13–16] algorithms. For combination approaches specifically, their first step aims at fog removal for a single image [12], which is the algorithm of low-level vision tasks in order to improve the performance of subsequent high-level vision tasks [17]; their second step aims at object detection, which in turn is a metric for the performance of the low-level task. Recently, deep learning methods for single image fog removal have achieved superior image quality; however, due to the network design features of subsequent high-level tasks not being jointly taken into account, combination models are often cumbersome and computationally inefficient. To overcome these problems, in this paper, we propose the joint semantic deep learning algorithm for object detection under foggy road conditions, referred to as $JSFR$.

The deep neural network of $JSFR$ is constructed by embedding attention mechanism modules (AM) and the $UNet$ decoding modules into the same scale corresponding submodules of the feature extraction module of object detection algorithm $FasterRCNN$. The main idea of our proposed algorithm can be shown intuitively in Figure 1. The novel algorithm improves the subsequent target detection performance through the joint representation of features before and after image recovery and is applied to detect seven types of objects on road traffics under foggy conditions.

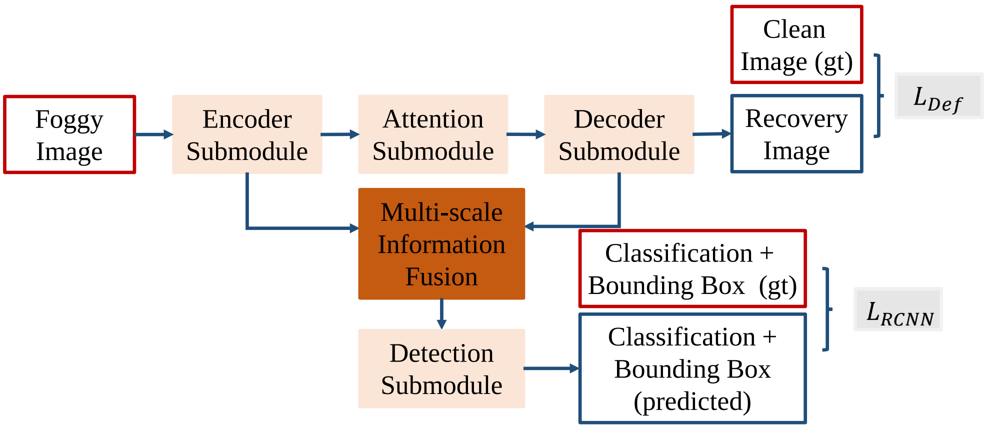
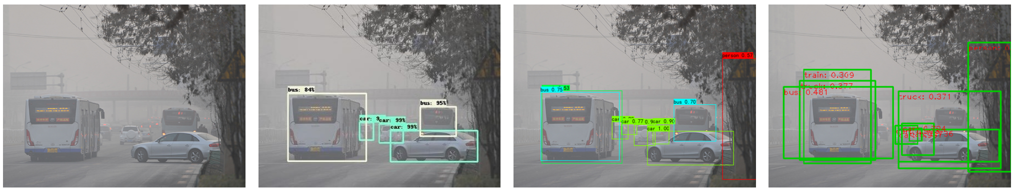


Figure 1. The idea of our proposed algorithm. The red boxes represent the inputs, the blue boxes represent the outputs, the pink blocks represent the submodules, the orange–red block represents the information fusion by element-wise addition, and the grey blocks represent the objective functions, which are defined in accordance with Equations (1) and (3) in the following text.

The main contributions are as follows:

- The algorithm is proposed for the joint tasks of low-level dehazing and high-level target detection. On the one hand, the two networks are combined to reduce the parameter scale and to improve the test time. On the other hand, the joint feature representations can help improve the robustness of road target detection under foggy or fog-free conditions. In addition, the joint network architecture is beneficial for improving the performances of both low-level and high-level tasks;
- The embedded attention mechanism module in the backbone network is conducive to capturing the position information of the fog at any time. Since fog is different from other noises, fog has the characteristics of fast drift and rapid position change, so determining the position information of fog before dehazing is beneficial for restoring image quality;
- Comprehensive experiments confirm the effectiveness of the proposed algorithm. Regardless of normal or foggy weather, and irrespective of synthetic or real data, the test results show the superiority of our algorithm. Figure 2 shows an example of its detection effect in the real world.



(**a**) Foggy Image (**b**) Ours (**c**) Faster R-CNN (**d**) DA

Figure 2. The detection effect in the real world. (**a**) indicates original image with fog; (**b**) indicates defogged image by our proposed method; (**c**) indicates defogged image by *Faster RCNN* [14]; (**d**) indicates defogged image by *DA* (Domain Adaptive) [6].

The remaining sections are arranged as follows: Section 2 reviews the *SOTA* object detection methods and single image fog removal methods; Section 3 gives an overview and introduces the formulation and optimization of our method; Comprehensive experiments are given in Section 4; and Section 5 concludes.

2. Related Work

In this section, deep neural network frameworks for object detection and for single image fog removal are reviewed in detail.

2.1. Deep Neural Network Based Object Detection Methods

Object detection is a fundamental task in computer vision. Many researchers developed various deep neural networks to improve the performance of object detection. These object detection methods can in general fall into two categories [18]: regression-based [13,16,19–21] and region-based suggestion methods [14,22,23].

Regression-based methods include: *Yolo* family [16,19,20], *SSD* [21] and *RetinaNet* [13], or their variants. The key idea is to use *CNN* modules to extract the images feature maps and then predict bounding box coordinates and category probabilities. For example, *Yolo* is first proposed by Redmon et al. [19] in 2016, which is constructed by 24 convolutional layers and 2 fully connected layers to achieve real-time detecting through end-to-end training. In the *Yolo* family, *Yolo V2* [20] and *Yolo V3* [16] obtain competitive detection performance by modifying the backbone network and adding a multi-scale prediction network module. Ref. [21] introduces *SSD*, which adds an auxiliary module to extract multi-scale feature maps and achieve object detection of different scales. *RetinaNet* achieves higher detection accuracy by introducing a focal loss function to balance the small sample category and reduce the running time in all one-stage object detection methods. Although the detection speed of the one-stage target detection algorithms is much higher than that of the two-stage target detection algorithms, the detection accuracy is much more inferior than the latter since the positioning of the target is not very accurate.

Region-based suggestion methods are two-stage object detection methods, including the *R-CNN* family [14,22,23]: *R-CNN* [22], *Fast RCNN* [23] and *Faster RCNN* [14], and *R-FNN* [24] and their variants. The key component of these combination methods is that a region-based suggestion module is proposed. In other words, the corresponding regions are extracted from the feature maps and then input into the *ROI* pooling module to generate fixed-length feature vectors for the classification and further regression of *bbox* (Bounding Box). It is worth mentioning that *Faster RCNN* has many advantages: it achieves the highest precision in all popular object detection methods; it can solve multi-scale and small target problems; it is easy to migrate to other categories and problems. So, *Faster RCNN* was chosen as the backbone network in this paper.

2.2. Object Detection under Foggy Weather Conditions

To address object detection under foggy weather, many researchers have constructed various deep learning methods from domain-adaptive or image recovery viewpoints, respectively.

Domain adaptive-based object detection methods under foggy weather conditions include [3–6,8] etc. In particular, Ref. [25] proposes a multi-level domain adaptive *Faster RCNN*, which uses different domain classifiers to supervise multi-scale feature alignment and improve the recognition ability by increasing domain classifiers. Ref. [26] proposes a robust multi-scale adversarial learning method for cross-domain object detection, which reduces image-level domain differences by multiple expansion convolution kernels, and reduces the influence of negative migration by excluding images and instances with low transfer ability. Ref. [27] proposes a stacked complementary loss method to achieve domain adaptation, which learns comprehensive discerning representations by detaching the gradient into several auxiliary losses in different network stages. Domain adaptation is a special case of transfer learning (*TL*). The idea is to project the features of different domains into the same feature space so that the target domain training can be enhanced by using data from other domains. It is a pity that *TL* has difficulty working well when there is a huge difference between two domains.

Another line of research focuses on image recovery. These strategies consist of embedding an additional image recovery module into the object detection backbone network [7,28], or recovering images first and then detecting objects for recovered images [11,12,29–31]. Ref. [28] proposes a dual subnet network (*DSNet*) to jointly learn three tasks including visibility enhancement, object classification, and object localization through embedding an image recovery module into *RetinaNet*. In [7], *Image-Adaptive YOLO* (*IA-YOLO*) is proposed, which configures a small convolutional neural network and a differentiable image processing module before *Yolo V3* to balance two tasks, i.e., image enhancement and object detection, under both foggy and low-light scenarios. Ref. [17] improves the detection performance by concatenating the dehazing module *AOD-Net* (All-in-One Dehazing Network) with the object detection network *Faster RCNN* by end-to-end training. Ref. [32] proposes a lightweight dehazing network *PDR-Net*.

The above-mentioned two groups of methods have greatly advanced the research of joint tasks including the low levels and high levels. However, the former often overlooks the image recovery potential information while the latter does not emphasize object detection performance and takes object detection as a task-driven evaluation index for image restoration.

Then a natural question arises about how to organically integrate the defogging and object detection modules to build an end-to-end joint tasks object detection algorithm under foggy weather. Can we design a network to complete the process of low-level image dehazing and high-level target recognition at the same system? In other words, can the network dealing with low-level vision tasks and the network dealing with high-level vision tasks be organically combined? Could it be possible not only to improve the generalization of subsequent high-level tasks, but also to save energy? To this end, the *JSFR* algorithm was proposed and applied to the detection of seven objects in road traffic scenarios.

3. Proposed Method

In this section, an overview of *JSFR* is first introduced in brief, and then the attention module and the parameters flow-chart of the noval fusion network are described in detail, and finally the formulation and the total loss function of *JSFR* are interpreted.

3.1. Framework Overview

For this paper, we designed *JSFR* to detect objects in road traffic under foggy weather conditions, as shown in Figure 3. The whole framework consists of four main modules: the image feature extraction module, the haze removal module, the multi-scale feature fusion module and the detection module. *Faster RCNN* embeds the four-level decoder of *UNet* [33] and the attention module [34] into the last, corresponding to the same scale sub-blocks of the feature extraction module. The haze removal module and the feature extraction module share weights to avoid additional computational burden. Finally, the four feature maps and the corresponding recovered feature maps are summed together and then input to the feature pyramid network (*FPN*) [35] to learn the multi-scale joint semantic representation, and then the output results are input to the detection module separately to improve the target detection accuracy under hazy weather conditions. The attention module is shown in Figure 4 and the setup of the fusion sub-module is shown in Figure 5.

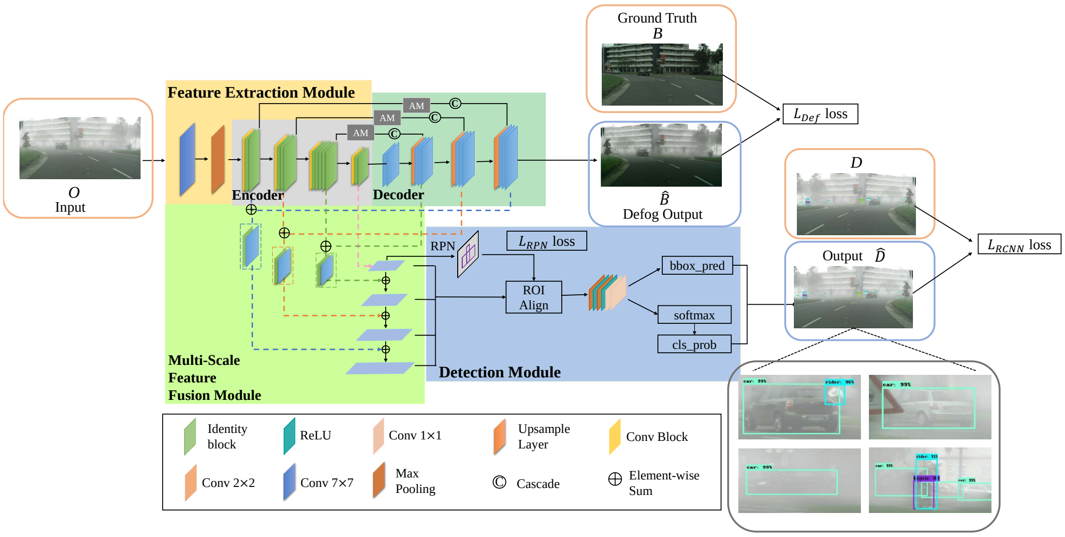


Figure 3. Framework overview for our *JSFR*. Input elements are marked by the orange boxes including the images O degraded by fog, the clean background images B, and ground truth images D labeled with class and bounding box. Output elements are marked by a blue box, including medium defog output $\hat{B}$ and final object detection results output $\hat{D}$. To see the results more clearly, we zoom in on the detection results for the final output image, as shown in the grey box. The orange/cyan/green/blue blocks represent the feature extraction module/UNet decoder/information fusion module/detection module, respectively.

3.2. Attention Module (AM) and Parameters Flow-Chart

Considering that different channel features of the image degraded by the fog represent different concentrations and different locations of the fog, *AM* modules [34] are embedded between the feature extraction module and *UNet* decoder, which is used to learn the foggy distribution. *AM* consists of channel and spatial attention, see Figure 4. The channel attention pays more attention to the concentration information of the fog in the input image, while the spatial attention module mainly focuses on the location information of the fog. The fog information can be captured adequately by the combination of the two modules.

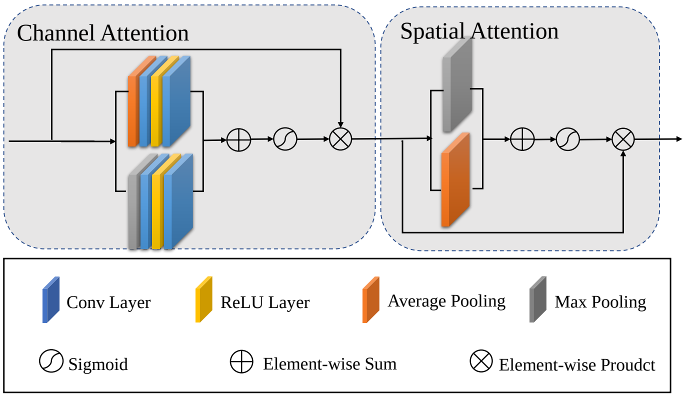


Figure 4. Attention Module framework [34].

3.3. The Parameters Flow-Chart of the Novel Fusion Network

In our model, the *UNet* encoder part is not designed separately, but shares the feature extraction layers with *Faster RCNN*, viz. *ResNet50*. We divide the 16 residual blocks of *ResNet50* into four residual block stages, denoted as *C2*, *C3*, *C4* and *C5*, respectively. The output of each residual block is embedded into the attention module separately and passed to the *UNet* Decoder part, and then its output is cascaded with the results of the feature extraction layers to fuse the low-level detail features and high-level semantic features to enrich the single image information. The whole parameters flow-chart is shown in Figure 5, which is consistent with *singl_u.py* in our all codes.

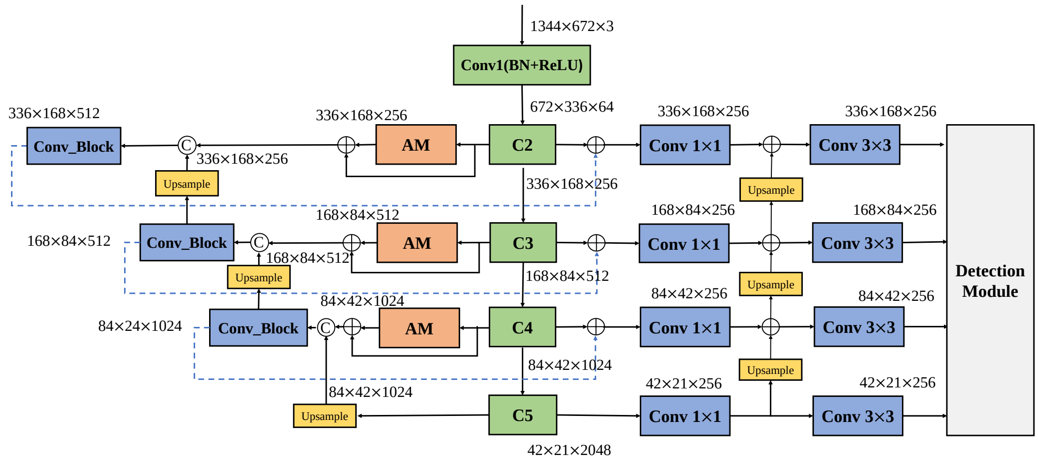


Figure 5. The parameters flow-chart of the novel fusion network. In the figure, the green blocks represent the feature extraction modules, the orange blocks represent the attention modules, the blue blocks represent the convolution blocks, the yellow blocks represent the upsampling, the grey block represents the detection module, © is for cascade, and ⊕ is element-wise addition.

3.4. Formalization and Loss Function

Let input image be $\mathcal{O} \in \mathcal{R}^{\,w\times h\times n}$, which is a tensor, where w and h are the length and width of the respective image, and n is the number of the channel. $f_1(\mathcal{O})$, $f_2(\mathcal{O})$, $f_3(\mathcal{O})$, $f_4(\mathcal{O})$ are obtained from the last four feature extraction sub-blocks of *Faster RCNN*. Secondly, $f_1(\mathcal{O})$, $f_2(\mathcal{O})$ and $f_3(\mathcal{O})$ are input into the output of the corresponding *AM* module and we obtain $h(f_1(\mathcal{O}))$, $h(f_2(\mathcal{O}))$, and $h(f_3(\mathcal{O}))$, respectively. Then, $f_4(\mathcal{O})$ is fed into the first layer of the *UNet* decoder and outputs $u(f_4(\mathcal{O}))$. Next, $u(f_4(\mathcal{O}))$ and $h(f_3(\mathcal{O}))$ are cascaded to pass the second layer of the *UNet* decoder, $\mathcal{B}_1 = u(h(f_3(\mathcal{O})); u(f_4(\mathcal{O})))$ is obtained. Similarly, we obtained successively $\mathcal{B}_2 = u(h(f_2(\mathcal{O})); \mathcal{B}_1)$; $\hat{\mathcal{B}} = u(h(f_2(\mathcal{O})); \mathcal{B}_2)$. So, the mean squared error (*MSE* loss) is used as one fine-tuning objective function L_{def} for recovering the quality of the foggy image, and L_{def} is operated by the equation:

$$L_{def} = \frac{1}{N}\sum_{i=1}^{N}\left\|\hat{B} - B\right\|^2, \tag{1}$$

where N is the number of foggy images.

Then, the element-wise sum of $\mathcal{B}_1$ and $f_3(\mathcal{O})$, $\mathcal{B}_2$ and $f_2(\mathcal{O})$, $\hat{\mathcal{B}}$ and $f_1(\mathcal{O})$ are sent to the corresponding scale sub-module of *FPN*, respectively. On the one hand, they are fused layer by layer from bottom to top in the *FPN* module and are outputted into the *RPN* network to generate regression detection boxes, and the detection boxes are classified into two categories (positive and negative). Therefore, the loss function of the *RPN* network is L_{rpn}:

$$L_{rpn} = -\frac{1}{N_{cls}}\sum_i \log\left[{p_i}^* p_i + \left(1-p_i^*\right)\left(1-p_i\right)\right] + \lambda\frac{1}{N_{reg}}\sum_i p_i^* \mathcal{L}_{1smooth}\left(t_i - t_i^*\right)$$

where,

$$\mathcal{L}_{1smooth} = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & otherwise\,, \end{cases} \tag{2}$$

where $t_i = \{t_x, t_y, t_w, t_h\}$ is a vector representing the four parameterized coordinates of the predicted *bbox*; t_i^* is the predicted vector′ coordinate; p_i is the probability of the predicted foreground and p_i^* is the probability of the predicted background; N_{cls} is the number of all objects.

On the other hand, the fused feature maps through FPN and the proposal's outputs through RPN are first added element-wise, and then sent into the ROI pooling layer. Next, these feature maps with ROI are fed into the subsequent fully connected layers, whose outputs are employed to classify seven objects and to estimate these objects' *bboxes*. Toward this purpose, we use the L_{rcnn} as a loss function as follows:

$$L_{rcnn} = \frac{1}{N_{cls}}\sum_{j=1}^{M} L_{cls}\left(p_j, p_j^*\right) + \lambda\frac{1}{N_{reg}}\sum_{j=1}^{M} p_j^* L_{reg}\left(t_j, t_j^*\right) \tag{3}$$

$$L = \alpha L_{def} + L_{rpn} + L_{rcnn}, \tag{4}$$

where t_j is the four coordinates of the label *bbox* of each object; t_j^* is the coordinate of its predicted *bbox*; M is the number of all objects, and M equals 7 in this paper; and $L_{reg}\left(t_j, t_j^*\right)$ is the $\mathcal{L}_{1smooth}$ function. p_j represents the probability of a label class, p_j^* represents the predicted probability of each class, and $L_{cls}\left(p_j, p_j^*\right)$ is the cross-entropy loss. The final network output is denoted as $\hat{D}$. The whole network is trained by L shown in Equation (4). It is noted that $\alpha \in [0,\ 1]$ is a hyperparameter to be fine-tuned to improve the performance of the object detection by recovering image quality. In ablation experiments, we set α equal to 1; in this way, the *mAP* of the proposed method achieves the highest value. See Section 4.4 for details.

4. Experimental Result

4.1. Experimental Setup

4.1.1. Implementation Details

To be fair, all algorithms were trained on the *Foggy Cityscapes* training subset again by end-to-end, and were tested on the synthetic and real images with fog or no-fog. *Foggy Cityscapes* was divided into a training subset, a validation subset and a testing subset at the ratio of 8:1:1. During training, we let the batchsize, epoch, confidence threshold be 2, 0.5, and 50, respectively. The initial learning rate was set to 5×10^{-4} and descent mode to segmented constant decay. All experiments were run on *Pytorch* with an *NVIDIA GeForce GTX3090*.

4.1.2. Evaluation Metric

Precision [36], *IoU* [19], *Recall* [36], *AP* [37], and *mAP* [28] were used to evaluate the object detection performance for our experimental results, whose expressions are reviewed in brief as follows:

$$IoU = \frac{A \cap B}{A \cup B}, \tag{5}$$

where *A* is the label of *bbox* and *B* is the predicted *bbox*. For each object, when the intersection over union ratio *IoU* is greater than 0.5, the detection is considered correct.

Additionally, precision, *recall* and *AP* are defined as:

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

$$AP = \int_0^1 P(R)dR, \tag{8}$$

where TP, FP, and FN are true positives, false positives, and false negatives, respectively, and $P/R \in [0,1]$ is Precision/Recall.

Finally, mAP is defined as

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i, \tag{9}$$

which is the average AP value over all object classes; N is the number of categories.

4.2. Datasets

Although there existed a few datasets for object detection under inclement weather, *Foggy Cityscapes* [38] was the synthetic dataset, while *Foggy Driving Dataset* [38], *RTTS* [39] and *Foggy Zurich Dataset* [40] were the real datasets. In this paper, *Foggy Cityscapes* was used to train these competing methods; *Foggy Cityscapes*, *Foggy Driving Dataset*, and *RTTS* were used to test these methods.

Foggy Cityscapes Dataset. *Foggy Cityscapes* contains eight categories, namely person, rider, car, truck, bus, train, motorcycle, and bicycle. As we were studying bus road traffic situations, we re-constructed the *Foggy Cityscapes* with seven categories composed of person, rider, car, truck, bus, motorcycle, and bicycle. In addition, considering that *Foggy Cityscapes* is based on *Cityscapes* [41], we could use images from *Cityscapes* as the corresponding ground truth (or label) from *Cityscapes*.

Real Datasets. The *Foggy Driving Dataset* [38] consists of 101 color images depicting real-world foggy driving scenes, which are used to test the performance of object detection in foggy weather. It consists of 509 vehicle instances (cars, trucks, buses, trains, motorcycles, and bicycles) and 290 human instances (people and riders), which are used for testing generalization. *RTTS* is also a real-world unpaired foggy image, which is a subset of *RESIDE* [39]. The format of *RTTS* is the same as that of *VOC2007*, and there exist five classes (cars, bicycles, motorcycles, people and buses) of objects with labels for object detection.

4.3. Experiment and Analysis

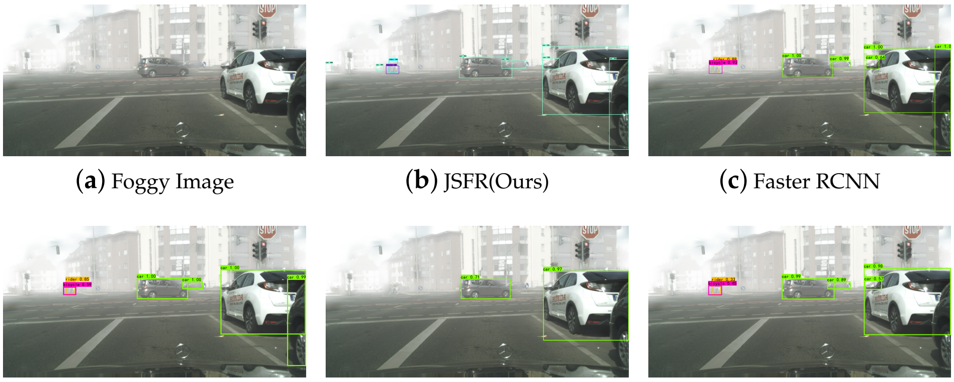
In this section, we compare the proposed algorithm *JSFR* with the popular object detection algorithms, the combination approaches, and the domain adaptive approaches by testing on synthetic or real images with and without fog.

4.3.1. Synthetic Foggy Image

To discuss the object detection performance under foggy weather, a qualitative comparison between our proposed *JSFR* with popular object detection methods *Faster RCNN* [14], *RetinaNet* [14], *Yolo V3* [16], *Efficient-Det* [15] is shown in Figures 6 and 7, and quantitative evaluations can be seen in Table 1. Figures 6 and 7 present scenes with sparse and dense fog, respectively. Regarding Figure 6, our algorithm recognizes seven vehicles, one rider and one bicycle. However, all others did not recognize the two distant vehicles. In addition, the confidence of all recognized objects by *JSFR* is not less than that of the others. While the detected object is a small target and in a dense foggy scene, as in Figure 7, our algorithm and *Faster RCNN* can identify the vehicles in the distance, but the confidence in recognized objects by *JSFR* is 0.99, which is higher than that of *Faster RCNN*. The quantitative results show that *JSFR* outperforms by up to 23%, 31.14%, 23.6% and 36% higher than *Faster RCNN*, *RetinaNet*, *Yolo V3*, and *Efficient-Det*, as shown in in Table 1.

Figures 8 and 9 and Table 2 compare the test results between *JSFR* and combinations such as *AODNet [17] + Faster RCNN*, *FFANet [42] + Faster RCNN*, *PSDNet [32] + Faster RCNN*, and domain adaptive based methods such as *DA* [6] and *ATF* [43]. Under the scene with sparse fog (refer to Figure 8) *JSFR* and *ATF* identified more targets than others, but there is still one person and a car not detected by *ATF* in Figure 8g. In the scene with dense fog, only *JSFR* recognized the small objects such as vehicles, while the others did not work on them; the confidence scores of all objects recognized by our method are higher than the corresponding scores of other methods (refer to Figure 9). From Table 2, we can see that the *mAP* of *JSFR* is up to 53.14%, which is higher than that of the others.

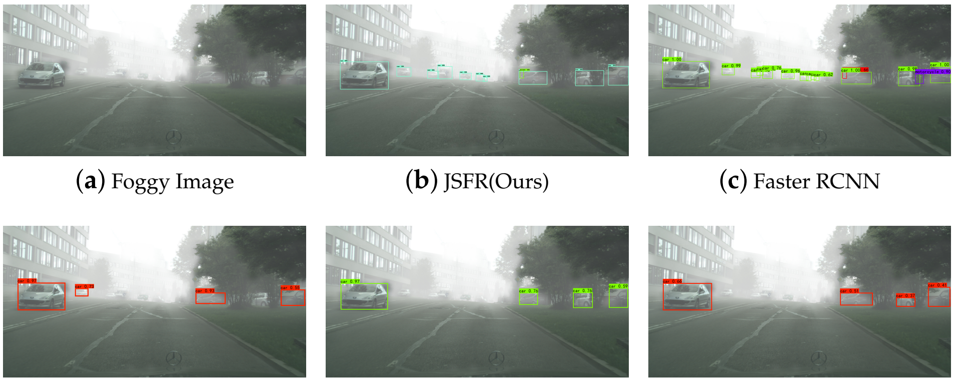
From Figures 6–9 and Tables 1 and 2, we summarize that the object detection performances of our algorithm, and domain adaptive and combined algorithms are all higher than those of the popular object detection algorithms under foggy weather, and our proposed method outperforms the others. So, we conclude that it is very necessary for the subsequent high-level object detection task that a sub-block is embedded into the whole object detection network to recover the quality from images degraded by fog.



(**a**) Foggy Image (**b**) JSFR(Ours) (**c**) Faster RCNN

(**d**) Yolo V3 (**e**) RetinaNet (**f**) Efficient-Det

Figure 6. Test results of *JSFU* and object detection methods including *Faster RCNN*, *Yolo V3*, *RetinaNet* and *Efficient-Det* on *Foggy Cityscapes* with sparse fog (electronic zoom-in recommended).



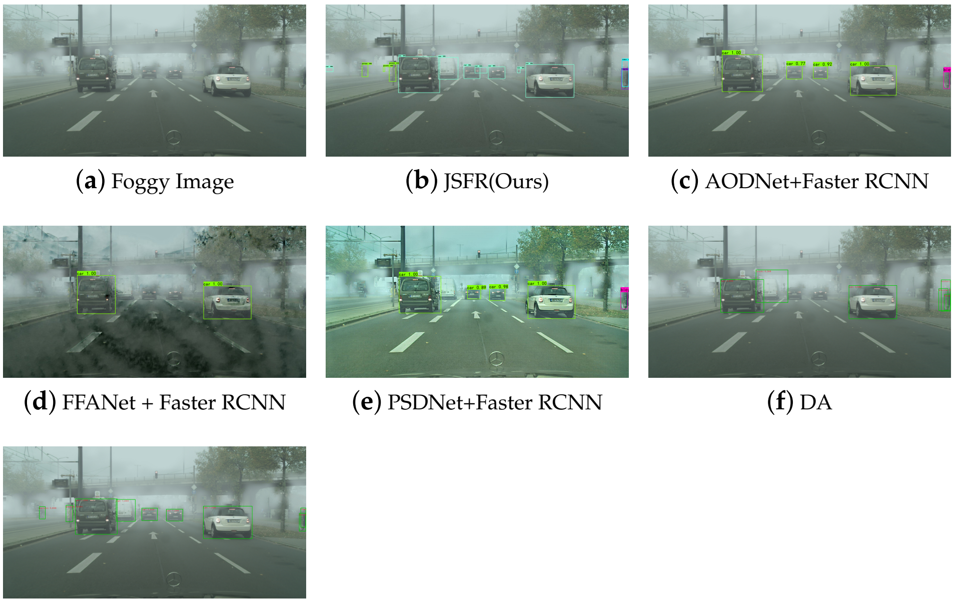
(**a**) Foggy Image (**b**) JSFR(Ours) (**c**) Faster RCNN

(**d**) Yolo V3 (**e**) RetinaNet (**f**) Efficient-Det

Figure 7. Testing results of *JSFU* and object detection methods including *Faster RCNN*, *Yolo V3*, *RetinaNet* and *Efficient-Det* on *Foggy Cityscapes* with dense fog (electronic zoom-in recommended).

Table 1. Comparison of *AP* and *mAP* between *JSFR* and object detection methods including *Faster RCNN*, *RetinaNet*, *Yolo V3*, and *Efficient-Det*.

Category	Faster RCNN	RetinaNet	Yolo V3	Efficient-Det	Ours
bicycle	0.36	0.28	0.36	0.23	**0.49**
bus	0.41	0.38	0.38	0.31	**0.51**
car	0.62	0.43	0.58	0.38	**0.74**
motorcycle	0.39	0.31	0.33	0.26	**0.43**
person	0.38	0.25	0.39	0.21	**0.58**
rider	0.44	0.32	0.43	0.26	**0.56**
truck	0.14	0.21	0.23	0.19	**0.41**
mAP	39.14%	31.14%	38.57%	37.86%	**53.14%**



(**a**) Foggy Image (**b**) JSFR(Ours) (**c**) AODNet+Faster RCNN

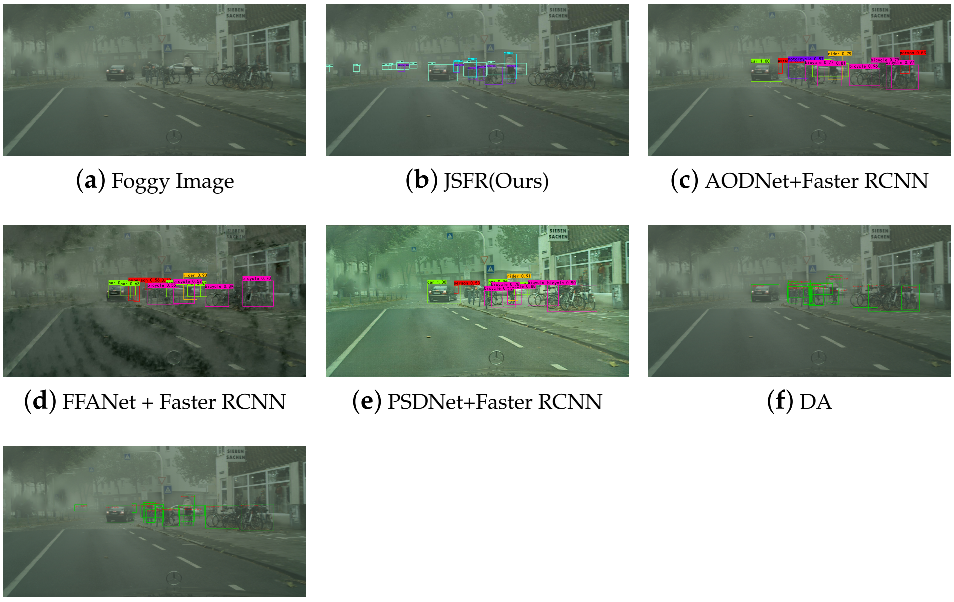
(**d**) FFANet + Faster RCNN (**e**) PSDNet+Faster RCNN (**f**) DA

(**g**) ATF

Figure 8. Results comparison of *Foggy Cityscapes* with sparse fog between *JSFU*, combination methods including *AODNet+Faster RCNN*, *FFANet + Faster RCNN*, *PSDNet + Faster RCNN*, and domain adaptive methods including *DA* and *ATF* (electronic zoom-in recommended).

Table 2. Comparison of *AP* and *mAP* between *JSFR*, combination methods including *AODNet+Faster RCNN*, *FFANet + Faster RCNN*, *PSDNet + Faster RCNN*, and domain adaptive methods including *DA* and *ATF*.

Category	AODNet	FFANet	PSDNet	DA	ATF	Ours
bicycle	0.27	0.24	0.32	0.45	0.39	**0.49**
bus	0.41	0.36	0.31	0.23	0.48	**0.51**
car	0.61	0.59	0.59	0.57	0.51	**0.74**
motorcycle	0.33	0.23	0.39	0.23	0.34	**0.43**
person	0.29	0.33	0.37	0.39	0.38	**0.58**
rider	0.38	0.37	0.45	0.46	0.48	**0.56**
truck	0.26	0.17	0.28	0.07	0.26	**0.41**
mAP	36.43%	32.71%	38.71%	32.85%	40.57%	**53.14%**



(**a**) Foggy Image

(**b**) JSFR(Ours)

(**c**) AODNet+Faster RCNN

(**d**) FFANet + Faster RCNN

(**e**) PSDNet+Faster RCNN

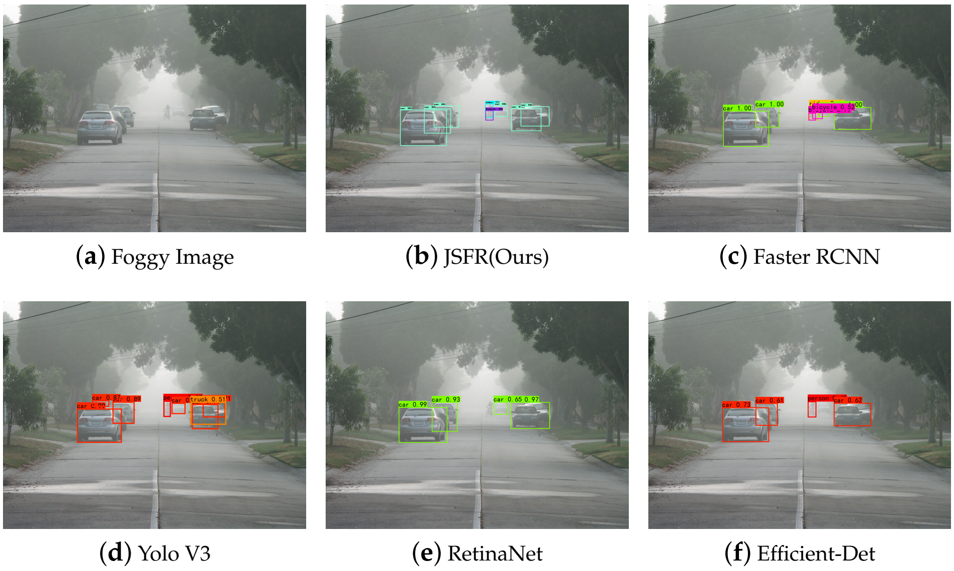
(**f**) DA

(**g**) ATF

Figure 9. Results comparison of *Foggy Cityscapes* with dense fog between *JSFU*, combination methods including *AODNet+Faster RCNN*, *FFANet + Faster RCNN*, *PSDNet + Faster RCNN*, and domain adaptive methods including *DA* and *ATF* (electronic zoom-in recommended).

4.3.2. Real Foggy Image

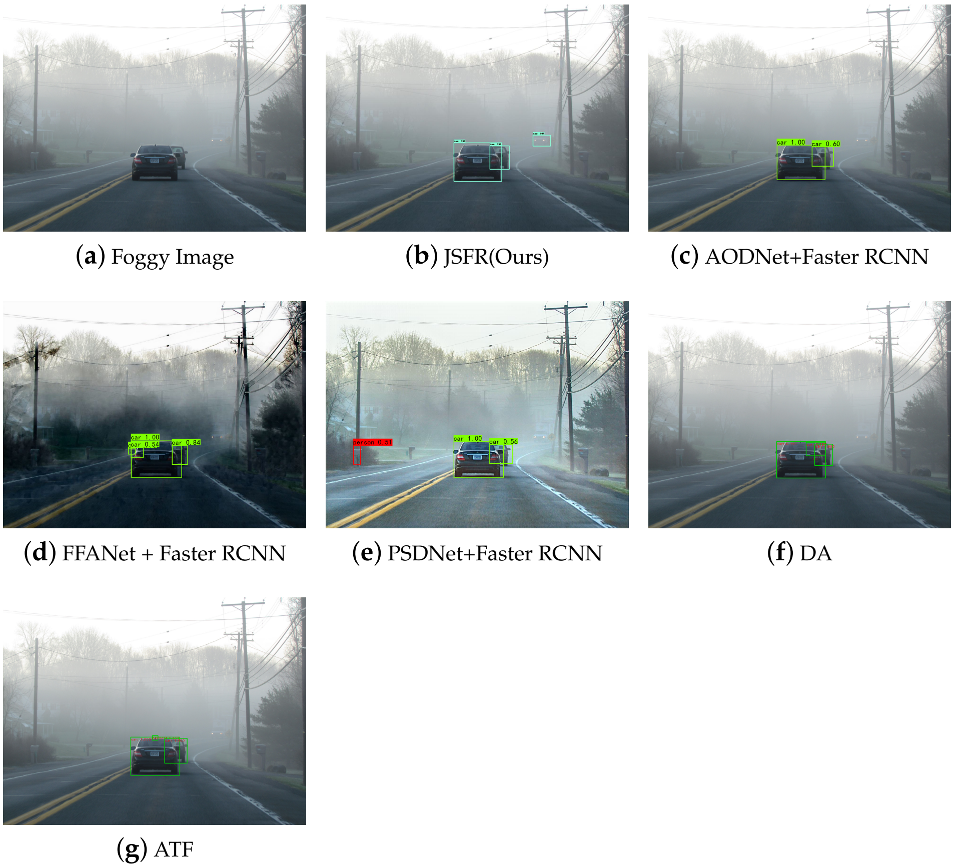
Figures 10 and 11 show the test results in real Foggy Driving Datasets. It can be concluded that the generalization of *JSFR* is better than that of others. For example, Figure 10b shows that our *JSFR* algorithm can correctly identify six cars, a rider, and a bicycle. However, *Yolo V3* can recognize six cars in Figure 10d, in which a car is mispredicted as a truck and a bicycle is not detected; *Efficient-Det* detected only three cars; only four cars were detected by *RetinaNet*. Referring to Figure 11, our algorithm detected a car at a distance in dense fog. The test results on *RTTS* are shown in Figures 12 and 13. In Figure 12, *JSFR*, *RetinaNet*, and *Yolo V3* detected the person with different confidence levels of 0.99, 0.97, and 0.52, respectively, so it is evident that our algorithm has the highest confidence. As seen in Figure 13, our algorithm had the ability to detect distant people which were not detected by other methods, and all confidence scores were also the highest.



(**a**) Foggy Image (**b**) JSFR(Ours) (**c**) Faster RCNN

(**d**) Yolo V3 (**e**) RetinaNet (**f**) Efficient-Det

Figure 10. Testing results of *JSFR* and object detection methods including *Faster RCNN*, *Yolo V3*, *RetinaNet*, and *Efficient-Det* on *Foggy Driving Dataset* (electronic zoom-in recommended).

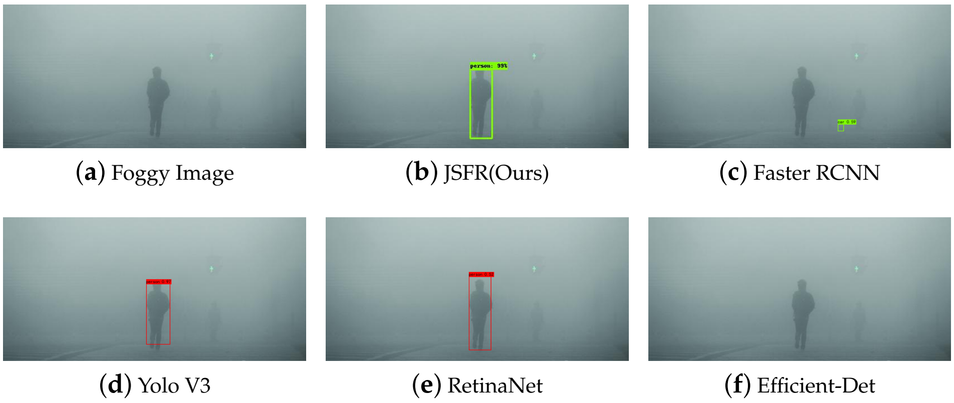


(**a**) Foggy Image (**b**) JSFR(Ours) (**c**) AODNet+Faster RCNN

(**d**) FFANet + Faster RCNN (**e**) PSDNet+Faster RCNN (**f**) DA

(**g**) ATF

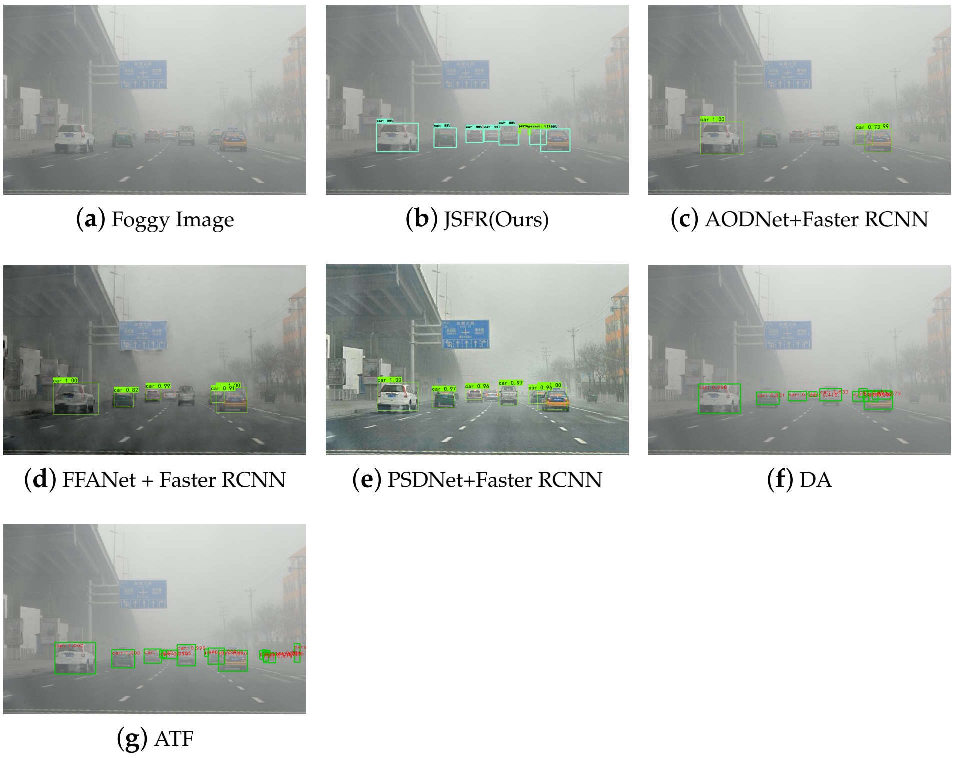
Figure 11. Testing results on *Foggy Driving Dataset* about *JSFR*, combination methods including *AODNet* + *Faster RCNN*, *FFANet* + *Faster RCNN*, and *PSDNet* + *Faster RCNN*, and domain adaptive methods including *DA* and *ATF*. (Electronic zoom-in recommended).



(**a**) Foggy Image (**b**) JSFR(Ours) (**c**) Faster RCNN

(**d**) Yolo V3 (**e**) RetinaNet (**f**) Efficient-Det

Figure 12. Testing results on *RTTS* for *JSFR* and object detection methods including *Faster RCNN*, *Yolo V3*, *RetinaNet*, and *Efficient-Det* (electronic zoom-in recommended).



(**a**) Foggy Image (**b**) JSFR(Ours) (**c**) AODNet+Faster RCNN

(**d**) FFANet + Faster RCNN (**e**) PSDNet+Faster RCNN (**f**) DA

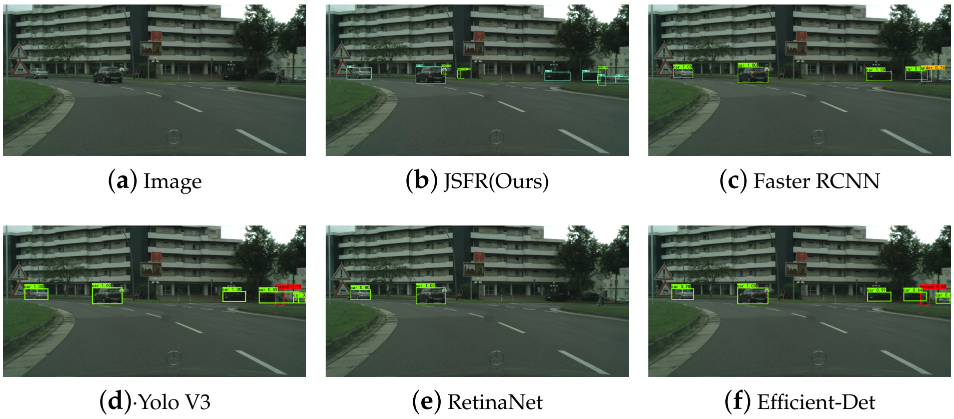
(**g**) ATF

Figure 13. Testing results on *RTTS* for *JSFR*, combination methods including *AODNet+Faster RCNN*, *FFANet + Faster RCNN*, and *PSDNet + Faster RCNN*, and domain adaptive methods including *DA* and *ATF* (electronic zoom-in recommended).

4.3.3. Synthetic Image Without Fog

In order to clarify that our method, *JSFR*, is also able to detect objects in normal weather, we performed a qualitative comparison and a quantitative evaluation by *mAP* between *JSFR* and *Faster RCNN*, *RetinaNet*, *Yolo V3*, *Efficient-Det* on *Cityscapes* [41].

As can be seen from Figure 14, *JSFR* can detect the person in the chair in normal weather, while the others cannot do it, and the confidence of all recognized targets is also the highest. Table 3 shows that the *mAP* of *JSFR* is higher than that of the other models, being 10.29%, 19.57%, 8.43% and 17.86% higher than *Faster RCNN*, *RetinaNet*, *Yolo V3* and *Efficient-Det*, respectively.


(**a**) Image (**b**) JSFR(Ours) (**c**) Faster RCNN

(**d**)·Yolo V3 (**e**) RetinaNet (**f**) Efficient-Det

Figure 14. Testing results on *Cityscapes* for *JSFR* and object detection methods including *Faster RCNN*, *Yolo V3*, *RetinaNet*, and *Efficient-Det* (electronic zoom-in recommended).

Table 3. Comparison of *AP* and *mAP* between *JSFR* and object detection methods including *Faster RCNN*, *RetinaNet*, *Yolo V3*, and *Efficient-Det* on *Cityscapes*.

Category	Faster RCNN	RetinaNet	Yolo V3	Efficient-Det	Ours
bicycle	0.31	0.24	0.33	0.24	**0.42**
bus	0.29	0.27	0.27	0.32	**0.26**
car	0.57	0.43	0.62	0.50	**0.73**
motorcycle	0.29	0.20	0.40	0.25	**0.31**
person	0.33	0.19	0.34	0.16	**0.54**
rider	0.35	0.17	0.34	0.12	**0.53**
truck	0.18	0.17	0.15	0.20	**0.25**
mAP	33.14%	28.86%	35.00%	25.57%	**43.43%**

4.4. Ablation Study

To verify the fine-tuning effect of the restored image module, we conducted an ablation experiment for hyperparameter α. Let α be 0, 1.0, and 0.5, respectively; we obtained four models such as *Faster RCNN*, *JSFR1*, *JSFR2*, and *JSFR*, which were tested on *Foggy Cityscapes*. From Table 4 it can be seen that the *mAP* values are the lowest if $\alpha = 0$ and the highest if $\alpha = 1$. So, we selected α equal to 1 in the final model, *JSFR1*.

Table 4. The ablation experiments for hyperparameter α.

α	Model	mAP
0	$Faster\ RCNN$	39.14%
0.1	$JSFR1$	50.39%
0.5	$JSFR2$	52.28%
1.0	$JSFR$	**53.14%**

4.5. Inference Time

We recorded the testing times of all models tested on *Foggy Cityscapes*, *RTTS*, *Foggy Driving Dataset* and *Cityscapes* in Table 5. An image from the four datasets can be tested in about 1.6 s by *JSFR* on a single *NVIDIA GeForce GTX 3090 GPU*. Our testing time is shorter than that of the combination and domain adaptive methods, while it is comparable to the time of the *Faster RCNN*. In other words, *JSFR* still maintains the testing speed of *Faster RCNN*, even though the *UNet* decoder module is embedded into *Faster RCNN*.

Table 5. Testing speeds of *JSFR*, combination methods including *AODNet+Faster RCNN*, *FFANet + Faster RCNN*, and *PSDNet + Faster RCNN*, domain adaptive methods including *DA* and *ATF*, and *Faster RCNN* on the four datasets with *GPU*.

Speed(s) Dataset Algorithm	Foggy Cityscapes	RTTS	Foggy Driving Dataset	Cityscapes
AODNet + Faster RCNN	2.68	2.55	2.54	2.68
FFANet + Faster RCNN	6.22	6.31	6.36	6.22
PSDNet + Faster RCNN	3.45	3.22	3.27	3.45
DA	2.73	2.66	2.65	2.73
ATF	2.71	2.56	2.54	2.72
Faster RCNN	1.66	1.56	1.56	1.66
JSFR	**1.69**	**1.54**	**1.55**	**1.69**

5. Conclusions

In this paper, *JSFR* is proposed to address multiple object detection in bus traffic road scene under foggy weather, which is designed to be a deep neural network by embedding *UNet* decoder and attention mechanism sub-modules into *Faster RCNN* for joint low-level and high-level task. Though *UNet* decoder achieves fog removal and recove the degraded image quality, its running time is still saved because of sharing parameters of the feature extracted module. The joint semantic representation is learned from recovered image and the image with fog by *FPN*, thus the fused information is leveraged to improve the the target detection performance. Comprehensive experiments confirm that *JSFR* outperforms other methods for target detection on synthetic and real image in both foggy and normal conditions. In the future, we will further improve the accuracy of the algorithm and reduce network parameters to meet the needs of landing implementation [44]. Secondly, we will continue to deeply study the application scenarios such as video detection of bus traffic roads and vehicle or pedestrian tracking. Thirdly, we will consider fusing overlap functions and fuzzy (rough) sets (see [45–49]) to design fog-attention-module. To provide convenience for researchers, all codes will be available at http://github.com/mendy-2013.

Author Contributions: Writing—original draft preparation, M.H.; Experiments and editing, Y.L.; review, J.F., writing—review, B.J. All authors have read and agreed to the published version of the manuscript.

Funding: This study was funded by the National Natural Science Foundation of China (No. 62071378), the Shaanxi Province International Science and Technology Cooperation Program (No. 2022KW-04), and the Xi'an Science and Technology Plan Project (No. 21XJZZ0072).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are openly available at [humingdi2005@github.com].

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

1. Hu, M.; Wu, Y.; Fan, J.; Jing, B. Joint Semantic Intelligent Detection of Vehicle Color under Rainy Conditions. *Mathematics* **2022**, *10*, 3512. [CrossRef]
2. Hu, M.; Wang, C.; Yang, J.; Wu, Y.; Fan, J.; Jing, B. Rain Rendering and Construction of Rain Vehicle Color-24 Dataset. *Mathematics* **2022**, *10*, 3210. [CrossRef]
3. Sindagi, V.; Oza, P.; Yasarla, R.; Patel, V. *Prior-Based Domain Adaptive Object Detection for Hazy and Rainy Conditions*; Springer: Cham, Switzerland, 2020; pp. 763–780.
4. VS, V.; Gupta, V.; Oza, P.; Sindagi, V.A.; Patel, V.M. MeGA-CDA: Memory Guided Attention for Category-Aware Unsupervised Domain Adaptive Object Detection. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 4514–4524.

5. Wang, T.; Zhang, X.; Yuan, L.; Feng, J. Few-Shot Adaptive Faster R-CNN. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 7166–7175.
6. Chen, Y.; Li, W.; Sakaridis, C.; Dai, D.; Van Gool, L. Domain Adaptive Faster R-CNN for Object Detection in the Wild. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3339–3348.
7. Liu, W.; Ren, G.; Yu, R.; Guo, S.; Zhu, J. Image-Adaptive YOLO for Object Detection in Adverse Weather Conditions. In Proceedings of the AAAI Conference on Artificial Intelligence, Arlington, TX, USA, 22 February–1 March 2022; pp. 1792–1800.
8. Xu, C.; Zhao, X.; Jin, X.; Wei, X. Exploring Categorical Regularization for Domain Adaptive Object Detection. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 11721–11730.
9. Liu, X.; Ma, Y.; Shi, Z.; Chen, J. GridDehazeNet: Attention-Based Multi-Scale Network for Image Dehazing. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 7313–7322.
10. Shao, Y.; Li, L.; Ren, W.; Gao, C.; Sang, N. Domain Adaptation for Image Dehazing. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 2805–2814.
11. Dong, H.; Pan, J.; Xiang, L.; Hu, Z.; Zhang, X.; Wang, F.; Yang, M.H. Multi-Scale Boosted Dehazing Network With Dense Feature Fusion. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 2154–2164.
12. Wu, H.; Qu, Y.; Lin, S.; Zhou, J.; Qiao, R.; Zhang, Z.; Xie, Y.; Ma, L. Contrastive Learning for Compact Single Image Dehazing. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021.
13. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 318–327. [CrossRef] [PubMed]
14. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
15. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 10778–10787.
16. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:abs/1804.02767.
17. Li, B.; Peng, X.; Wang, Z.; Xu, J.; Feng, D. AOD-Net: All-in-One Dehazing Network. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 4780–4788.
18. Zhao, Z.; Zheng, P.; Xu, S.; Wu, X. Object Detection With Deep Learning: A Review. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3212–3232. [CrossRef] [PubMed]
19. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
20. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.
21. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2015.
22. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
23. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
24. Dai, J.; Li, Y.; He, K.; Sun, J. R-FCN: Object Detection via Region-based Fully Convolutional Networks. *arXiv* **2016**, arXiv:abs/1605.06409.
25. Xie, R.; Yu, F.; Wang, J.; Wang, Y.; Zhang, L. Multi-Level Domain Adaptive Learning for Cross-Domain Detection. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Republic of Korea, 27–28 October 2019; pp. 3213–3219.
26. Pan, Y.; Ma, A.J.; Gao, Y.; Wang, J.; Lin, Y. Multi-Scale Adversarial Cross-Domain Detection with Robust Discriminative Learning. In Proceedings of the 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), Snowmass Village, CO, USA, 1–5 March 2020; pp. 1313–1321.
27. Shen, Z.; Maheshwari, H.; Yao, W.; Savvides, M. SCL: Towards Accurate Domain Adaptive Object Detection via Gradient Detach Based Stacked Complementary Losses. *arXiv* **2019**, arXiv:abs/1911.02559.
28. Huang, S.; Le, T.; Jaw, D. DSNet: Joint Semantic Learning for Object Detection in Inclement Weather Conditions. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 2623–2633. [CrossRef] [PubMed]
29. Zhang, F.; Li, Y.; You, S.; Fu, Y. Learning Temporal Consistency for Low Light Video Enhancement from Single Images. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 19–25 June 2021; pp. 4965–4974.

30. Ma, L.; Ma, T.; Liu, R.; Fan, X.; Luo, Z. Toward Fast, Flexible, and Robust Low-Light Image Enhancement. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 5627–5636.
31. Lamba, M.; Mitra, K. Restoring Extremely Dark Images in Real Time. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 19–25 June 2021; pp. 3486–3496.
32. Chen, Z.; Wang, Y.; Yang, Y.; Liu, D. PSD: Principled Synthetic-to-Real Dehazing Guided by Physical Priors. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 7176–7185.
33. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention*; Springer: Cham, Switzerland, 2015; pp. 234–241.
34. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
35. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944.
36. Jose, H.; Vadivukarasi, T.; Devakumar, J. Extraction of Protein Interaction Data: A Comparative Analysis of Methods in Use. *Eurasip J. Bioinform. Syst. Biol.* **2007**, *2007*, 53096. [CrossRef] [PubMed]
37. Hu, M.; Bai, L.; Fan, J.; Zhao, S.R.; Chen, E. Vehicle Color Recognition Based on Smooth Modulation Neural Network with Multi-scale Feature Fusion. *Front. Comput. Sci.* **2022**, *17*, 173321. [CrossRef]
38. Sakaridis, C.; Dai, D.; Van Gool, L. Semantic Foggy Scene Understanding with Synthetic Data. *Int. J. Comput. Vis.* **2018**, *126*, 973–992. [CrossRef]
39. Li, B.; Ren, W.; Fu, D.; Tao, D.; Feng, D.; Zeng, W.; Wang, Z. Benchmarking Single-Image Dehazing and Beyond. *IEEE Trans. Image Process.* **2019**, *28*, 492–505. [CrossRef] [PubMed]
40. Sakaridis, C.; Dai, D.; Hecker, S.; Van Gool, L. Model Adaptation with Synthetic and Real Data for Semantic Dense Foggy Scene Understanding. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 707–724.
41. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The Cityscapes Dataset for Semantic Urban Scene Understanding. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 3213–3223.
42. Qin, X.; Wang, Z.; Bai, Y.; Xie, X.; Jia, H. FFA-Net: Feature Fusion Attention Network for Single Image Dehazing. In Proceedings of the AAAI Conference on Artificial Intelligence, Atlanta, GA, USA, 8–12 October 2019.
43. He, Z.; Zhang, L. Domain Adaptive Object Detection via Asymmetric Tri-Way Faster-RCNN. In *Computer Vision ECCV 2018*; Springer: Cham, Switzerland, 2020; pp. 309–324.
44. Hu, M.; Yang, J.; Lin, N.; Liu, Y.; Fan, J. Lightweight single image deraining algorithm incorporating visual saliency. *IET Image Process.* **2022**, *16*, 3190–3200. [CrossRef]
45. Wang, J.; Zhang, X. A novel multi-criteria decision-making method based on rough sets and fuzzy measures. *Axioms* **2022**, *11*, 275. [CrossRef]
46. Liang, R.; Zhang, X. Pseudo general overlap functions and weak inflationary pseudo BL-algebras. *Mathematics* **2022**, *10*, 3007. [CrossRef]
47. Zhang, X.; Liang, R.; Bustince, H.; Bedregal, B.; Fernandez, J.; Li, M.; Ou, Q. Pseudo overlap functions, fuzzy implications and pseudo grouping functions with applications. *Axioms* **2022**, *11*, 593. [CrossRef]
48. Sheng, N.; Zhang, X. Regular partial residuated lattices and their filters. *Mathematics* **2022**, *10*, 2429. [CrossRef]
49. Wang, J.; Zhang, X.; Hu, Q. Three-way fuzzy sets and their applications (II). *Axioms* **2022**, *under review of the second version*.