Object Detection in Foggy Scenes by Embedding Depth and Reconstruction into Domain Adaptation

Xin Yang[1[0000−0002−4617−5733]], Michael Bi Mi[2], Yuan Yuan[2], Xin Wang[2], and
Robby T. Tan[1,3[0000−0001−7532−6919]]

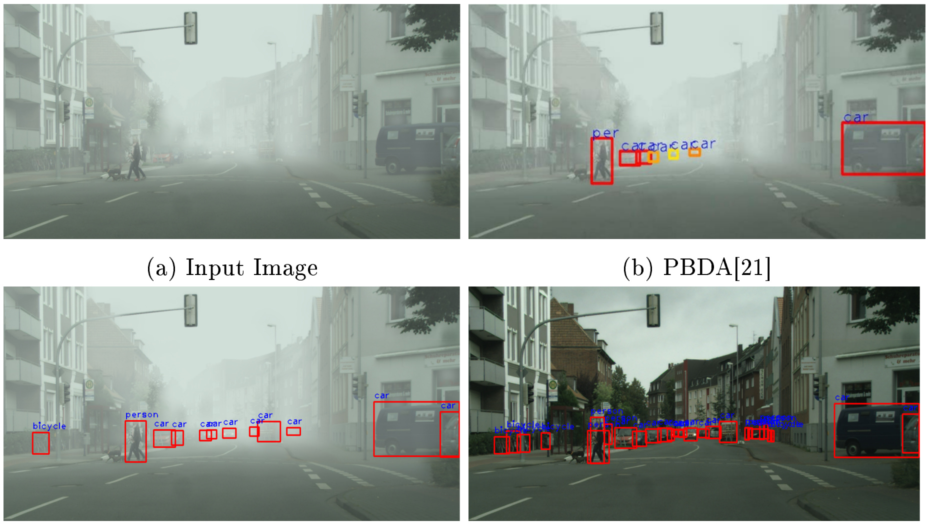
[1] National University of Singapore
[2] Huawei International Pte Ltd
[3] Yale-NUS College
e0674612@u.nus.edu, michaelbimi@yahoo.com,
{yuanyuan10,wangxin237}@huawei.com, robby.tan@{nus,yale-nus}.edu.sg

Abstract. Most existing domain adaptation (DA) methods align the features based on the domain feature distributions and ignore aspects related to fog, background and target objects, rendering suboptimal performance. In our DA framework, we retain the depth and background information during the domain feature alignment. A consistency loss between the generated depth and fog transmission map is introduced to strengthen the retention of the depth information in the aligned features. To address false object features potentially generated during the DA process, we propose an encoder-decoder framework to reconstruct the fog-free background image. This reconstruction loss also reinforces the encoder, i.e., our DA backbone, to minimize false object features. Moreover, we involve our target data in training both our DA module and our detection module in a semi-supervised manner, so that our detection module is also exposed to the unlabeled target data, the type of data used in the testing stage. Using these ideas, our method significantly outperforms the state-of-the-art method (47.6 mAP against the 44.3 mAP on the Foggy Cityscapes dataset), and obtains the best performance on multiple real-image public datasets. Code is available at: https://github.com/VIML-CVDL/Object-Detection-in-Foggy-Scenes

Keywords: Domain adaptation · Object detection · Foggy scenes.

1 Introduction

Object detection is impaired by bad weather conditions, particularly fog or haze. Addressing this problem is important, since many computer vision applications, such as self-driving cars and video surveillance, rely on robust object detection regardless of the weather conditions. One possible solution is to employ a preprocessing method, such as image defogging or dehazing [25, 14] right before an object detection module. However, this solution is suboptimal, since bad weather image enhancement itself is still an open problem, and thus introduces a risk of removing or altering some target object information.



(a) Input Image

(b) PBDA[21]

(c) **Our Result**

(d) Ground-Truth

Fig. 1: (a) Input dense fog image from Foggy Cityscapes [5]. (b) Result from PBDA [21], where many objects are undetected. (c) Ours, where more objects are detected. (d) Ground-truth, annotated from its corresponding clear image. Zoom in for better visualization.

Recently, object detection methods based on unsupervised domain adaptation (DA) (e.g., [7, 3, 4]) have shown promising performance for bad weather conditions. By aligning the source (clear weather image) and the target (weather degraded image) distributions in the feature level, a domain adaptive network is expected to produce weather-invariant features. Unlike image pre-processing methods, domain adaptive detection networks do not require an additional defogging module during the inference stage, and can also work on both clear and foggy conditions. The DA methods, however, were not initially proposed for the adverse weather conditions, and hence align the source and target features based only on feature alignment losses, ignoring some important aspects of the target data, such as depth, transmission map, reconstruction of object instances of the target data, etc. This is despite the fact that for bad weather, particularly fog or haze, these aspects can be imposed. Sindagi et al. [21] attempt to fuse DA with adverse weather physics models, but do not obtain a satisfactory performance (39.3 mAP on Foggy Cityscapes, compared to 47.6 mAP of ours).

In this paper, we propose a DA method that learns domain invariant features by considering depth cues, consistency between depth from fog transmission, and clear-image reconstruction. Moreover, we involve the source and target data to train our whole network in a semi-supervised manner, so that our object detection module can be exposed to the unlabeled target data, which has the same type as data in the testing stage. Most existing DA methods aim to suppress any discrepancies between the source and target data in the feature space, and this includes any depth distribution discrepancies. However, depth information is critical in object detection [6, 23, 17], and thus the depth suppression will significantly affect the object detection performance. To resolve this, we propose a depth estimation module and its corresponding depth loss, so that the depth information in our features can be retained. This depth loss forces our DA backbone to retain the depth information during the DA process. Moreover, to further reinforce our DA backbone to retain the depth information, we also add a transmission-depth consistency loss.

When performing DA, the source and target images are likely to contain different object instances, and aligning two different objects with different appearances encourages the generation of false features. To address this problem, we fuse a reconstruction module into our DA backbone, and propose a reconstruction loss. Based on the features from our DA backbone's layers from the target (fog image), our reconstruction module generates a clear image. Our reconstruction loss thus measures the difference between the estimated clear image and the clear-image pseudo ground-truth of the target obtained from an existing defogging method. This reconstruction loss will then prevent false features generated during the feature extraction process. During the training stage, the DA model gradually becomes more robust to fog, and the predictions on the unlabeled target images become more reliable. This gives us an opportunity to employ the target data to train our object detection module, so that the module can be exposed to both source and target data and becomes less biased to the source data.

Fig. 1 shows our object detection result, which incorporates all our losses and ideas into our DA backbone. As a summary, our contributions and novelties are as follow:

- Without imposing our depth losses, DA features are deprived from the depth information, due to the over-emphasis on source/target adaptation. This deprivation negatively affects object detection performance. Hence, we introduce depth losses to our DA backbone to retain the depth information in our DA features.
- We propose to reinforce the target transmission map to have consistent depth information to its corresponding depth estimation. This consistency loss constraints the transmission map and improves further the DA performance and the depth retention.
- We propose to integrate an image reconstruction module into our DA backbone. Hence, any additional false object features existing in DA features will be penalized, and hence minimized.

Our quantitative evaluations show that our method outperforms the state-of-the-art methods on various public datasets, including real image datasets.

2 Related Work

Object Detection in Foggy Scenes Most existing object detection models require a fully-supervised training strategy [28]. However, under adverse weather conditions, having sufficient images and precise annotations is intractable. A possible solution is to utilize defogging algorithms. The defogged images are less affected by fog, and hence they can be fed into object detection models which are trained with clear images directly. However, defogging is still an existing research problem and thus limits the performance potential of this approach. Moreover, defogging introduces an additional computational overhead, hindering the real-time process for some applications. These drawbacks were discussed and analyzed in [19, 21, 26].

Domain Adaptation DA methods were proposed to train a single network which can work on different domains. In DA, there will be labeled images from the source domain and unlabeled images from the the target domain. During the training stage, images from both domains will be fed into the network. The source images with annotations will be used for the training of object detection part. Meanwhile, a domain discriminator will examine which domain the extracted feature maps come from. The discriminator will be rewarded for accurate domain prediction, but the network will be penalized. Hence, the network is encouraged to extract domain invariant feature maps, i.e., fog free features. Since the feature maps are already domain invariant, the detection trained with the source labels can also detect objects from the target images. Additionally, once the network is trained, the domain discriminator is not needed anymore, hence no additional overhead in the inference stage.

There are a few existing DA methods that tried to perform DA between clear weather and fog weather. Some methods investigated where and how to put the domain discriminators, so that the DA can be more efficient whilst retain most object-related information [3, 18, 20, 4, 26]. [21, 15, 10, 22, 12, 27] aimed to designed a more suitable domain discriminator, using transmission maps, entropy, uncertainty masks, memory banks/dictionaries and class clusters. However, most of these methods focus on synthetic datasets, and ignores the fact where the weather-specific knowledge prior can also be integrated to better describe the domain discrepancy.

3 Proposed Method

Fig. 2 shows the pipeline of our method, where clear images are our source input, and foggy images are our target input. For the source images, we have their corresponding annotations (bounding boxes and classes) to train our object detection module. For the target images, we do not have any annotations. In this DA framework, we introduce a few constraints: depth, consistency between the transmission and depth, and clear background reconstruction. The goal of adding these constraints is to extract features from both source (clear image) and target (fog image) that are robust for object detection. Moreover, we exploit the target

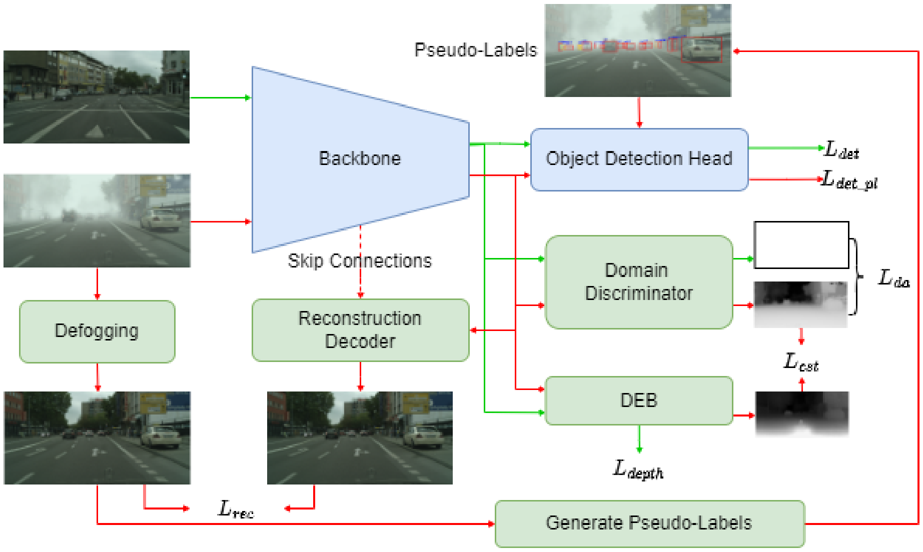


Fig. 2: The network consists of five parts. (1) Backbone extracts feature maps from the input images. (2) Object detection head localizes and categorizes object instances from the feature maps. (3) Domain discriminator and DEB encourage the backbone to extract fog-invariant features, and maintains the images' depth distributions. (4) Reconstruction decoder minimizes the fake object features generated by DA. (5) Pseudo-Labels involve target domain information in the pipeline, and apply consistency regularization between fog and defogged images. The green arrows represent source data-flow, and the red arrows represent target data-flow. Note, only the blue modules are needed in the testing stage.

predictions to train our object detection module, so that the module can be exposed to the unlabeled target data, and hence less biased to the source data.
Object Detection For the object detection module, we employ Faster-RCNN [16], which consists of a backbone $\mathcal{F}$ for feature extraction, and an object detection head, $\mathcal{G}$. The loss for the object detection is defined as:

$$\mathcal{L}_{\text{det}} = \mathcal{L}_{\text{rpn}} + \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{bbox}}, \tag{1}$$

where, $\mathcal{L}_{\text{rpn}}$ is the regional proposal loss, $\mathcal{L}_{\text{cls}}$ is the classification loss, and $\mathcal{L}_{\text{bbox}}$ is the localization loss.
Domain Adaptation Our domain adaptation backbone shares the same backbone as that of the object detection module. For the domain discriminator, we use transmission maps as the domain indicator (i.e., the discriminator is expected to produce a blank map for source, and a transmission map for target). The corresponding loss can be defined as:

$$\mathcal{L}_{\text{da}} = \|\mathcal{D}(\mathcal{F}(I_s))\|_2^2 + \|t - \mathcal{D}(\mathcal{F}(I_t))\|_2^2, \tag{2}$$

where, $\mathcal{F}$ is the backbone, $\mathcal{D}$ is the domain discriminator. I_s and I_t are the input images from the source domain and the target domain, respectively. t is the transmission map for the target image.

3.1 Depth Estimation Block (DEB)

In the DA process, it is unlikely that a pair of source and target images to have the same depth distribution, since they unlikely contain the same scenes. Thus, when the existing DA methods suppress the domain discrepancies, the depth information is also suppressed in the process. However, recent methods have shown that the depth information benefits object detection [6, 23, 17], which implies that suppression of depth can affect the performance of object detection.

To address this problem, we need to retain the depth information during the DA feature alignment. We introduce Depth Estimation Block (DEB), a block that generates a depth map based on the extracted features from our DA backbone. We define the depth recovery loss as follows:

$$\mathcal{L}_{\text{depth}} = \|\text{DEB}(\mathcal{F}(I_s)) - D_{gt}\|_2^2\,, \tag{3}$$

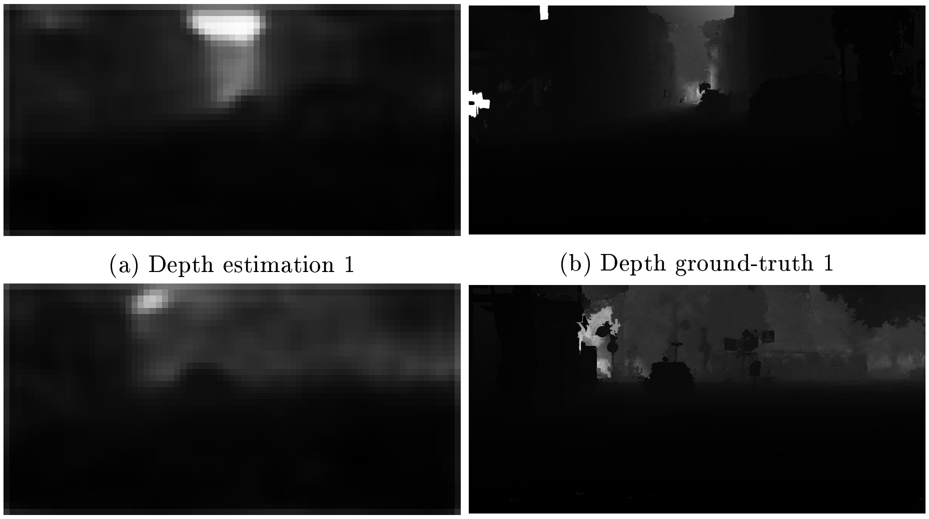
where, D_{gt} is the ground-truth depth map, which is resized to the same size as the corresponding feature map. $\mathcal{F}(I_s)$ represents the source feature maps, and DEB() is our DEB module. For datasets such as Cityscapes [5], they provide the ground-truth depth maps. For the other datasets which do not provide depth ground-truth, we need to generate the depth maps as a pseudo ground-truth using the existing depth estimation networks, such as [8, 2, 9]. Note, DEB is only trained on the source images. Unlike the transmission DA loss in Eq. (2), we backpropagate the depth loss over both our DA backbone and DEB to retain the depth information in our DA features. Fig. 3 shows our depth estimations. Note that our goal here is not to have accurate depth estimation, but to retain depth cues in our features.

3.2 Transmission-Depth Consistency

In foggy scenes, we can model the transmission of light throughout the fog particles as $t = \exp(-\beta D)$, where t is the transmission, D is the depth, and β as the fog particles attenuation factor. As one can notice, there is a strong correlation between the transmission and depth. Hence, we reinforce our predicted transmission and depth to be consistent:

$$\mathcal{L}_{\text{cst}} = \|Norm(-\log(\mathcal{D}(\mathcal{F}(I_t)))) - Norm(\text{DEB}(\mathcal{F}(I_t)))\|_2^2\,, \tag{4}$$

where $\mathcal{D}(\mathcal{F}(I_t))$ is the generated transmission map from the domain discriminator, and $\text{DEB}(\mathcal{F}(I_t))$ represents the estimated depth map. $Norm()$ represents a normalization operator. Like most defogging methods, we assume that β is uniform across the input target. Since the transmission and the depth values are the same only up to a scale, we normalize their values, and thus consequently cancel out β in the process. This consistency loss enforces the consistency between the depth encoded in the estimated transmission and the depth from our DEB, this constraint leads to more robust depth information in our features.



(a) Depth estimation 1

(b) Depth ground-truth 1

(c) Depth estimation 2

(d) Depth ground-truth 2

Fig. 3: Examples of the estimated depths in comparison with the ground-truths. Note that, the estimated depths look blurry as they have a low resolution (38 × 75). We can observe that the estimated depth maps (a) and (c) from DEB match the depth distribution patterns in the ground-truth depth maps (b) and (d), respectively, which indicates that our feature maps retain the depth information.

3.3 Reconstruction DA

Since DA methods use unpaired source and target that most likely contain different object instances, when the DA backbone aligns the features, the alignment will occur on two different object instances (e.g., car in the source, and motorbike in the target). Hence, when the DA backbone is suppressing such discrepancy, consequently it can generate false features, which can harm the object detection performance.

To address this problem, we regularize the generation of object features by fusing a reconstruction decoder into our DA backbone. This decoder reconstructs the features back to the clear background image. To train this decoder, we use either the clear image or the defogged image of the target image as the ground-truths. Since the reconstruction ground-truths have the same object instances as in the target images, the reconstruction loss will prevent our DA backbone from generating false instance features. Our reconstruction loss is defined as:

$$\mathcal{L}_{\text{rec}} = \left\|\mathcal{R}(\mathcal{F}(I_t)) - I^{de}\right\|_2^2, \tag{5}$$

where $\mathcal{R}$ is the reconstruction module and $\mathcal{R}(\mathcal{F}(I_t))$ represents the reconstructed target image. I^{de} is the clear/defogged target image, which we use as the ground-truth for the reconstruction. Only target images are involved in the reconstruction, as there are no fog distortions in the source images.

3.4 Learning from Target

In many DA cases, target data does not have ground-truths, and hence [21, 3, 4, 26, 18, 20, 15, 10, 22] use the source domain's knowledge to train the object detection module, right after the domain feature adaptation process. This means the object detection module is never exposed to the target data, which is likely to be different from the source data in terms of the appearances of the object instances (e.g., the shapes of road signs in one country are different from those of the road signs from another country, etc.). The main problem why many methods do not use target data in training the detection module is because the target data is unlabeled.

With the help from our DA module, our detection module's performance on the target data is improving over iterations. As the predictions on target become more accurate, we can select some reliable predictions as pseudo-labels to train our detection module. Hence, our training is split into two stages for each iteration: In the first stage, we generate pseudo-labels from the whole network; in the second stage, we do DA training involving the generated pseudo-labels. To obtain more reliable predictions in the first stage, we employ a defogging method to augment our target input. The augmented target input in the form of defogged image will enable our network to estimate the bounding boxes and class labels. If the network has high confidence with these estimates, it means the estimates can be considered as reliable and used as pseudo-labels. In the second stage, we feed the same target input to our network without augmentation, and let it to predict the bounding boxes and class labels. We then enforce these network's estimates to be consistent with the pseudo-labels. This process encourages the network to become more robust to fog, and to expose our object detection module to the target data. Note that, in the end of each iteration, we employ Exponential Moving Average (EMA) in our network to generate more reliable predictions.

Total Loss Combining all the losses we introduced above, we can derive our overall loss as:

$$\mathcal{L} = \mathcal{L}_{\text{det}} + \lambda\mathcal{L}_{\text{da}} + a\mathcal{L}_{\text{depth}} + b\mathcal{L}_{\text{cst}} + c\mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{det\_pl}}, \tag{6}$$

where, λ, a, b, c are the weight parameters to control the importance of the losses. $\mathcal{L}_{\text{det\_pl}}$ is the detection loss with pseudo-labels. Note that, in the testing stage, we only use our DA backbone and the detection module. In other words, all the additional modules (i.e., the domain discriminator, the depth estimation block, the reconstruction module or the pseudo-labels) does not affect the runtime in the testing stage.

4 Experimental Results

We compare our DA method with recent DA methods: [21, 3, 4, 26, 18, 20, 15, 10, 22, 12, 27], where the last two are published as recent as this year. To make the comparison fair, we use the same base of object detection, which is Faster-RCNN [24]. For the backbone, our method uses a pretrained ResNet-101 [24]. We set the confidence threshold τ to be 0.8 for all the experiments. More details of our blocks can be found in the supplementary material. Our overall network is trained end-to-end. We follow the same training settings as in [21, 3, 4, 18, 20], where the networks are trained for 60K iterations, with a learning rate of 0.002. We decrease the learning rate by a factor of 10 for every 20K iterations. The weights parameters λ, a, b, c are empirically set to be 0.1, 10, 1, 1, respectively.

As for the datasets, the Cityscapes dataset is a real world street scene dataset provided by [5], and all images were taken under clear weather. Based on this dataset, [19] simulates synthetic fog on each clear image, and creates the Foggy Cityscapes dataset. We use the same DA settings as in [21, 3, 4, 18, 20], where 2975 clear images and 2975 foggy images are used for training, and 495 foggy images are used for evaluation.

Aside from the Cityscapes dataset, STF (Seeing Through Fog) [1], Foggy-Driving [19], and RTTS [13] are the datasets with real world foggy images used in our experiments. STF dataset categories its images into different weather conditions, we choose *clear weather daytime* as our source domain and *fog daytime* as our target domain. We randomly select 100 images from *fog daytime* as our evaluation set, and use the rest to train the network. For RTTS, we follow the same DA settings as in [21, 20]. For FoggyDriving, it only contains 101 fog images, which is insufficient for DA training. Hence, we evaluate the DA models trained on Cityscapes/Foggy Cityscapes directly on these datasets.

4.1 Quantitative Results

The synthetic Foggy Cityscapes dataset has the ground-truth transmission maps, depth maps and the clear background of the target images for reconstruction, thus we can use the ground-truths directly in our training process, however for fair comparisons we do not use them. Instead we employ DCP [11] to compute the transmission maps, reconstruction maps, and use it as the defogging pre-processing module when involving target predictions. As for the depth, we employ Monodepth [8] to compute the pseudo ground-truths. We also employ DCP and Monodepth for real data that have no ground-truths of clear images, transmission maps, and depth maps. Note that, the methods we use to generate pseudo ground-truths (DCP and Monodepth) are not the state-of-the-art methods, as we want to show that our DA's performances are not limited by the precision of the pseudo ground-truths.

The results on this dataset are provided in Table 1. The mAP threshold for all the models is 0.5. When comparing DA models, there are two important non-DA baseline models that need to be considered. One is the model trained on clear images but tested on foggy images, which we call Lowerbound. Any DA models should performance better than this Lowerbound model. In our experiment, Lowerbound is 28.12 mAP for Foggy Cityscapes dataset. The other model is both trained and tested on clear images, which we call Upperbound. Since it is not affected by fog at all, the goal of DA models is to approach its performance, but it is not possible to exceed it. In our experiment, Upperbound is 50.08 mAP

Table 1: Quantitative results of Ours compared to the existing DA methods evaluated against Foggy Cityscapes testing set. AP (%) of each category and the mAP (%) of all the classes. Bold numbers are the best scores, and underlined numbers are the second best scores. Our mAP outperforms the best existing method over 3%.

s

Method		Backbone	person	rider	car	truck	bus	train	motor	bicycle	mAP
Baseline	Faster-RCNN	ResNet-101	32.0	39.5	36.2	19.4	32.1	9.4	23.3	33.2	28.1
DA Methods	DA-Faster[3]	ResNet-101	37.2	46.8	49.9	28.2	42.3	30.9	32.8	40.0	38.5
	SWDA[18]	VGG-16	29.9	42.3	43.5	24.5	36.2	32.6	30.0	35.3	34.3
	SCL[20]	ResNet-101	30.7	44.1	44.3	30.0	47.9	42.9	29.6	33.7	37.9
	PBDA[21]	ResNet-152	34.9	46.4	51.4	29.2	46.3	43.2	31.7	37.0	40.0
	MEAA[15]	ResNet-101	34.2	48.9	52.4	30.3	42.7	46.0	33.2	36.2	40.5
	UaDAN[10]	ResNet-50	36.5	46.1	53.6	28.9	49.4	42.7	32.3	38.9	41.1
	Mega-CDA[22]	VGG-16	37.7	49.0	52.4	25.4	49.2	46.9	34.5	39.0	41.8
	SADA[4]	ResNet-50-FPN	**48.5**	**52.6**	**62.1**	29.5	50.3	31.5	32.4	**45.4**	44.0
	CaCo[12]	VGG-16	38.3	46.7	48.1	33.2	45.9	37.6	31.0	33.0	39.2
	MGA[27]	VGG-16	43.9	49.6	60.6	29.6	50.7	39.0	38.3	42.8	44.3
	Ours	ResNet-101	39.9	51.6	59.0	**39.7**	**58.0**	**49.1**	**39.2**	45.1	**47.6**

Table 2: Quantitative results of Ours compared to the existing DA methods evaluated against the STF testing set. AP (%) of each category and the mAP (%).

Method		PassengerCar	LargeVehicle	RidableVehicle	mAP
Baseline	Faster-RCNN	74.0	54.1	25.7	51.3
DA Methods	DA-Faster	77.9	51.5	25.2	51.6
	SWDA	77.2	50.1	24.5	50.6
	SCL	78.1	52.5	20.7	50.4
	SADA	**78.5**	52.2	24.2	51.6
	Ours	**78.5**	**57.4**	**30.2**	**55.4**

for Foggy Cityscapes dataset. Table 1 shows that our proposed method performs better than any other DA methods.

For the real world datasets, we cannot compute Lowerbound's and Upperbound's performance, since we do not have the clear background of the foggy images. Thus, we can only compare our models with the performance of the other DA methods. The results are presented in Tables. 2 to 4. Our model achieved a better performance on all the real world datasets. For FoggyDriving, we can observe that our model trained on synthetic dataset can also generalize well on the real world image datasets. Note that, the compared methods are not as many as the previous table, since some methods only provided their performances on the synthetic datasets, and we do not have their data or code to evaluate on the real world datasets.

Table 3: Quantitative results of Ours compared to the existing DA methods evaluated against the FoggyDriving testing set. mAP (%) of all the categories.

Method		mAP
Baseline	FRCNN	26.41
DA Methods	DA-Faster	31.60
	SWDA	33.54
	SCL	33.49
	SADA	32.26
	Ours	**34.62**

Table 4: Quantitative results of Ours compared to the existing DA methods evaluated against the RTTS testing set. AP (%) of each category and the mAP (%).

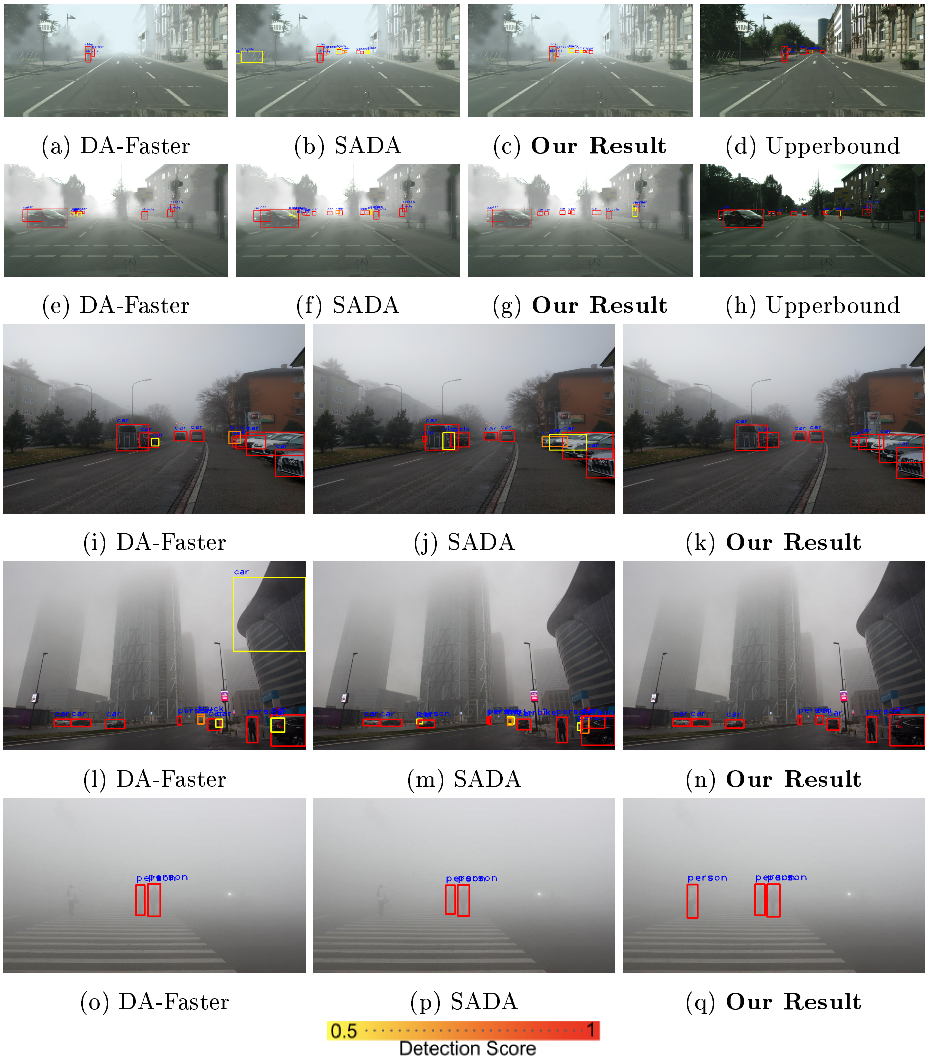
Method		person	car	bus	motor	bicycle	mAP
Baseline	Faster-RCNN	46.6	39.8	11.7	19.0	37.0	30.9
DA Methods	DA-Faster	42.5	43.7	16.0	18.3	32.8	30.7
	SWDA	40.1	44.2	16.6	23.2	41.3	33.1
	SCL	33.5	48.1	18.2	15.0	28.9	28.7
	PBDA	37.4	**54.7**	17.2	22.5	38.5	34.1
	SADA	37.9	52.7	14.5	16.1	26.2	29.5
	Ours	**47.7**	53.4	**19.1**	**30.2**	**49.3**	**39.9**

4.2 Qualitative Results

The qualitative results are presented in Fig. 4. We evaluate our model on both synthetic and real world datasets, and compare it with DA-Faster[3] and SADA[4]. For the synthetic dataset, we also compare the model with Upperbound to visualize how close their predictions are. Note again that, Upperbound is the Faster-RCNN model trained on the clear training set and tested on the clear testing set. We can see that our method can detect more objects compared to DA-Faster. Both SADA and our method can detect most of the object instances in fog. However, we can see that SADA generated some false predictions. Our method removed some false predictions, and thus the final object detection performance is approaching Upperbound.

4.3 Ablation Studies

Table 5 shows the ablation studies on Foggy Cityscapes to demonstrate the importance of each loss. The check mark indicates which losses are involved. In the table, DA represents the performance with domain discriminator only, DEB represents the depth recovery loss, Consist represents the transmission-depth consistency loss. Reconst represents the reconstruction loss using the pseudo



(a) DA-Faster (b) SADA (c) **Our Result** (d) Upperbound

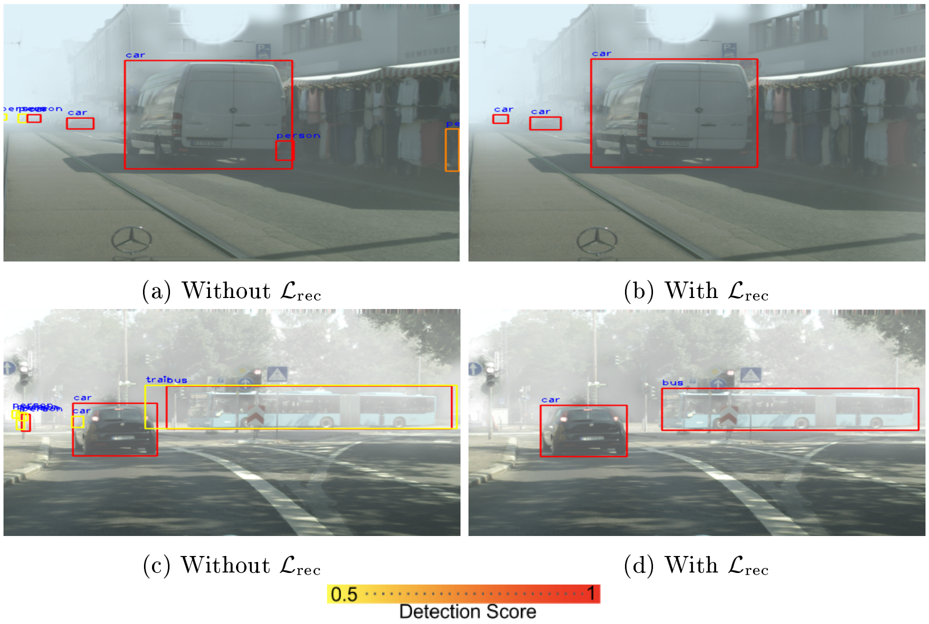
(e) DA-Faster (f) SADA (g) **Our Result** (h) Upperbound

(i) DA-Faster (j) SADA (k) **Our Result**

(l) DA-Faster (m) SADA (n) **Our Result**

(o) DA-Faster (p) SADA (q) **Our Result**

Fig. 4: Comparisons with DA-Faster[3], SADA[4], and Upperbound. The first two rows are the comparison on Cityscapes → Foggy Cityscapes. Our model detects more objects and reduces false positive predictions and approaches the Upperbound performance. The last three rows are the comparison on real world images. Our model has more true positive detections and less false positive detections. Different bounding box's colour represents a different confidence score.



(a) Without $\mathcal{L}_{\text{rec}}$

(b) With $\mathcal{L}_{\text{rec}}$

(c) Without $\mathcal{L}_{\text{rec}}$

(d) With $\mathcal{L}_{\text{rec}}$

Fig. 5: Ablation studies on our reconstruction loss. (a)(c): False positive predictions when there is no reconstruction, from model (DA+DEB+Consist). (b)(d): After we added our reconstruction loss. The false positive predictions are reduced.

ground-truth images generated from DCP [11]. We also provide Fig. 5 to show the comparisons between the models with and without our reconstruction module. PL means we include target predictions as pseudo labels for training. As one can notice, each of the losses improves the overall mAP. The network has a performance gain with only pseudo ground-truths of transmission maps, depth maps, and defogged images. This once again shows that our method works without the need of precise ground-truths. If we use the ground-truths of transmission maps, depth maps, and defogged images (which are actually available for Foggy Cityscape), our performance reaches overall 49.2 mAP. The weights of the losses are chosen to ensure that all the losses will contribute to the training properly. If we set a from 10 to 1, or we set b from 1 to 0.1, the performance drops by around 1 mAP. If we set c from 1 to 0.1, the performance drops by 2 mAP. Setting λ to be 0.1 is recommended by a few DA papers. The performance drops below 40 mAP if λ becomes too large. In our method, the weights are set empirically.

5 Conclusion

We have proposed a novel DA method with a reconstruction as a regularization, to develop an object detection network which is robust for fog or haze conditions.

Table 5: Ablation study of our model against the Foggy Cityscapes testing set. mAP is used to analyze the effectiveness of each loss.

DA	DEB	Consist	Reconst	PL	mAP
✓					42.6
✓	✓				43.3
✓	✓	✓			45.3
✓	✓	✓	✓		45.8
✓	✓	✓	✓	✓	47.6

To address the problem that DA process can suppress depth information, we proposed the DEB to recover it. We proposed the transmission-depth consistency loss to reinforce the transmission map based DA to follow the target image's depth distribution. We integrated a reconstruction module to our DA backbone to reconstruct a clear image of the target image and reduce the false object instance features. We involved target domain knowledge into DA, by reusing reliable target predictions and enforcing consistent detection. We evaluated the framework on several benchmark datasets showing that our method outperforms the state-of-the-art DA methods.

References

1. Bijelic, M., Gruber, T., Mannan, F., Kraus, F., Ritter, W., Dietmayer, K., Heide, F.: Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11682–11692 (2020)
2. Chen, P.Y., Liu, A.H., Liu, Y.C., Wang, Y.C.F.: Towards scene understanding: Unsupervised monocular depth estimation with semantic-aware representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2624–2632 (2019)
3. 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 IEEE conference on computer vision and pattern recognition. pp. 3339–3348 (2018)
4. Chen, Y., Wang, H., Li, W., Sakaridis, C., Dai, D., Van Gool, L.: Scale-aware domain adaptive faster r-cnn. International Journal of Computer Vision **129**(7), 2223–2243 (2021)
5. 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 IEEE conference on computer vision and pattern recognition. pp. 3213–3223 (2016)
6. Ding, M., Huo, Y., Yi, H., Wang, Z., Shi, J., Lu, Z., Luo, P.: Learning depth-guided convolutions for monocular 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 1000–1001 (2020)
7. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: International conference on machine learning. pp. 1180–1189. PMLR (2015)

8. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 270–279 (2017)
9. Godard, C., Mac Aodha, O., Firman, M., Brostow, G.J.: Digging into self-supervised monocular depth estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3828–3838 (2019)
10. Guan, D., Huang, J., Xiao, A., Lu, S., Cao, Y.: Uncertainty-aware unsupervised domain adaptation in object detection. IEEE Transactions on Multimedia (2021)
11. He, K., Sun, J., Tang, X.: Single image haze removal using dark channel prior. IEEE transactions on pattern analysis and machine intelligence **33**(12), 2341–2353 (2010)
12. Huang, J., Guan, D., Xiao, A., Lu, S., Shao, L.: Category contrast for unsupervised domain adaptation in visual tasks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1203–1214 (2022)
13. Li, B., Ren, W., Fu, D., Tao, D., Feng, D., Zeng, W., Wang, Z.: Benchmarking single-image dehazing and beyond. IEEE Transactions on Image Processing **28**(1), 492–505 (2018)
14. Liu, X., Ma, Y., Shi, Z., Chen, J.: Griddehazenet: Attention-based multi-scale network for image dehazing. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7314–7323 (2019)
15. Nguyen, D.K., Tseng, W.L., Shuai, H.H.: Domain-adaptive object detection via uncertainty-aware distribution alignment. In: Proceedings of the 28th ACM International Conference on Multimedia. pp. 2499–2507 (2020)
16. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28**, 91–99 (2015)
17. Saha, S., Obukhov, A., Paudel, D.P., Kanakis, M., Chen, Y., Georgoulis, S., Van Gool, L.: Learning to relate depth and semantics for unsupervised domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8197–8207 (2021)
18. Saito, K., Ushiku, Y., Harada, T., Saenko, K.: Strong-weak distribution alignment for adaptive object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6956–6965 (2019)
19. Sakaridis, C., Dai, D., Van Gool, L.: Semantic foggy scene understanding with synthetic data. International Journal of Computer Vision **126**(9), 973–992 (2018)
20. Shen, Z., Maheshwari, H., Yao, W., Savvides, M.: Scl: Towards accurate domain adaptive object detection via gradient detach based stacked complementary losses. arXiv preprint arXiv:1911.02559 (2019)
21. Sindagi, V.A., Oza, P., Yasarla, R., Patel, V.M.: Prior-based domain adaptive object detection for hazy and rainy conditions. In: European Conference on Computer Vision. pp. 763–780. Springer (2020)
22. 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 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4516–4526 (2021)
23. Vu, T.H., Jain, H., Bucher, M., Cord, M., Pérez, P.: Dada: Depth-aware domain adaptation in semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7364–7373 (2019)
24. Yang, J., Lu, J., Batra, D., Parikh, D.: A faster pytorch implementation of faster r-cnn. https://github.com/jwyang/faster-rcnn.pytorch (2017)

25. Zhang, H., Patel, V.M.: Densely connected pyramid dehazing network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3194–3203 (2018)
26. Zhou, Q., Gu, Q., Pang, J., Feng, Z., Cheng, G., Lu, X., Shi, J., Ma, L.: Self-adversarial disentangling for specific domain adaptation. arXiv preprint arXiv:2108.03553 (2021)
27. Zhou, W., Du, D., Zhang, L., Luo, T., Wu, Y.: Multi-granularity alignment domain adaptation for object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9581–9590 (2022)
28. Zou, Z., Shi, Z., Guo, Y., Ye, J.: Object detection in 20 years: A survey. arXiv preprint arXiv:1905.05055 (2019)