Every Pixel Matters: Center-aware Feature Alignment for Domain Adaptive Object Detector

Cheng-Chun Hsu[1], Yi-Hsuan Tsai[2], Yen-Yu Lin[1,3], and Ming-Hsuan Yang[4,5]

[1]Academia Sinica [2]NEC Labs America
[3]National Chiao Tung University [4]UC Merced [5]Google Research

Abstract. A domain adaptive object detector aims to adapt itself to unseen domains that may contain variations of object appearance, viewpoints or backgrounds. Most existing methods adopt feature alignment either on the image level or instance level. However, image-level alignment on global features may tangle foreground/background pixels at the same time, while instance-level alignment using proposals may suffer from the background noise. Different from existing solutions, we propose a domain adaptation framework that accounts for each pixel via predicting pixel-wise objectness and centerness. Specifically, the proposed method carries out center-aware alignment by paying more attention to foreground pixels, hence achieving better adaptation across domains. We demonstrate our method on numerous adaptation settings with extensive experimental results and show favorable performance against existing state-of-the-art algorithms. Source codes and models are available at https://github.com/chengchunhsu/EveryPixelMatters.

1 Introduction

As a key component to image analysis and scene understanding, object detection is essential to many high-level vision applications such as instance segmentation [4,10,11,12], image captioning [38,20,37], and object tracking [18]. Although significant progress on object detection [9,29,26] had been made, an object detector that can adapt itself to variations of object appearance, viewpoints, and backgrounds [2] is always in demand. For example, a detector used for autonomous driving is required to work well under diverse weather conditions, even if training data may be acquired under some particular weather conditions.

To address this challenge, *unsupervised domain adaptation* (UDA) methods [28,7,36,31,35] have been developed to adapt models trained on an annotated source domain to another unlabeled target domain. Adopting a similar strategy to the classification task [36] using adversarial feature alignment, numerous UDA methods for objection detection [14,32,16,1,21,22,2,15] are proposed to reduce the domain gap across source and target domains. However, such alignment is usually performed on the image level that adapts global features, which is less effective when the domain gap is large [32,5]. To improve upon global alignment[1],

[1] In this paper, we use image-level alignment and global alignment interchangeably.

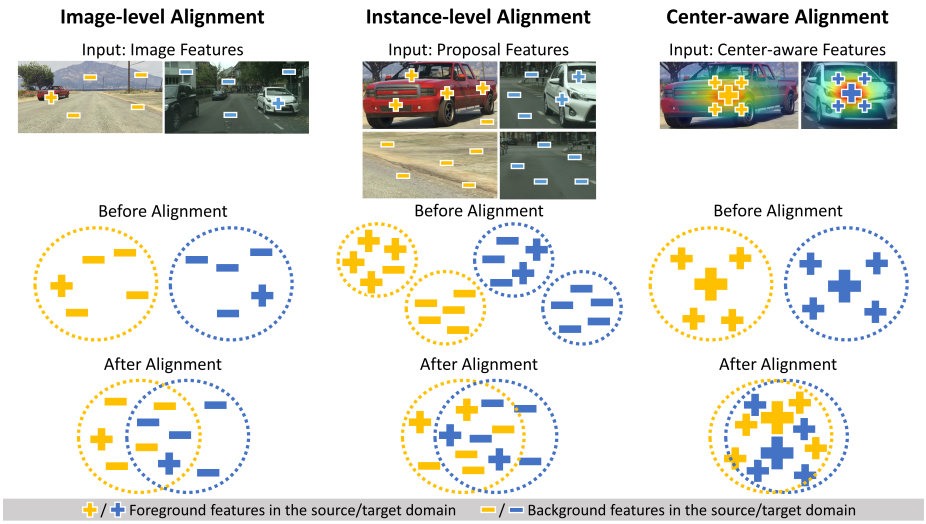


Fig. 1. Comparisons between different alignment methods. 1) For image-level alignment, it considers both foreground/background pixels, which may lead to noisy alignment and focus more on background pixels. 2) Instance-level alignment is performed on proposals, in which the pooled feature on all the pixels within the proposal could mix foreground/background signals. In addition, proposals in the target domain may contain much more background pixels due to the domain gap. 3) The proposed center-aware alignment focuses on foreground pixels with higher confidence scores of objectness and centerness, i.e., those marked by larger "+" showing higher centerness response, which play a crucial role to reduce the confusion during alignment.

existing methods [2,14,41] adapt instance-level distributions that pool features of all the pixels within a proposal. However, since pixel distributions are unknown in the target domain, the proposal extracted from the target domain could contain many background pixels. As a result, this may significantly confuse the alignment procedure when adapting instance-level features of target proposals to the source distribution that contains mostly foreground pixels (Fig. 1).

In this paper, we propose to take every pixel into consideration when aligning feature distributions across two domains. To this end, we design a module to estimate pixel-wise objectness and centerness of the entire image, which allows our alignment process to focus on foreground pixels, instead of the proposal that may contain tangled foreground/background pixels as considered in the prior work. In order to predict the pixel-wise information, we revisit the object detection framework and adopt fully-convolutional layers. As a result, our method aims to align the centered discriminative part of the objects across domains, namely the regions with high objectness scores and close to the object centers (see Fig. 1). Thereby, these regions are less sensitive to irrelevant background pixels in the target domain and facilitate distribution alignment. To the best of

our knowledge, we make the first attempt to leverage pixel-wise objectness and centerness for domain adaptive object detection.

To validate the proposed method, we conduct extensive experiments on three benchmark settings for domain adaptation: Cityscapes [3] → Cityscapes Foggy [33], Sim10k [17] → Cityscapes, and KITTI [8] → Cityscapes. The experimental results show that our center-aware feature alignment performs favorably against existing state-of-the-art algorithms. Furthermore, we provide ablation study to demonstrate the usefulness of each component in our method. The major contributions of this paper are summarized as follows. First, we propose to discover discriminative object parts on the pixel level and better handle the domain adaptation task for object detection. Second, center-aware distribution alignment with its multi-scale extension is presented to account for object scales and alleviate the unfavorable effects caused by cluttered backgrounds during adaptation. Third, comprehensive ablation studies validate the effectiveness of the proposed framework with center-aware feature alignment.

2 Related Work

In this section, we review a few research topics relevant to this work, including object detection and domain adaptive object detection.

2.1 Object Detection

Object detection studies can be categorized into anchor-based and anchor-free detectors. Anchor-based detectors compile a set of anchors to generate object proposals, and formulate object detection as a series of classification tasks over the proposals. Faster-RCNN [30] is the pioneering anchor-based detector, where the region proposal network (RPN) is employed for proposal generation. Owing to its effectiveness, RPN is widely adopted in many anchor-based detectors [25,26].

Anchor-free detectors skip proposal generation, and directly localize objects based on the fully convolutional network (FCN) [27]. Recently, anchor-free methods [23,40,6] leverage keypoint (i.e., the center or corners of a box) localization and achieve comparable performance with anchor-based methods. Yet, these methods require complex post-processing for grouping the detected points. To avoid such a process, FCOS [34] proposes per-pixel prediction, and directly predicts the class and offset of the corresponding object at each location on the feature map. In this work, we take advantages of the property in anchor-free methods to identify discriminate areas for the alignment procedure.

2.2 UDA for Object Detector

Chen *et al.* [2] first present two alignment practices, *i.e.*, image-level and instance-level alignments, by adopting adversarial learning at image and instance scales, respectively. For image-level alignment, Saito *et al.* [32] further indicate that

Table 1. Alignment schemes adopted by existing methods, including global alignment (G), instance-level alignment (I), low-level feature alignment (L), pixel-level alignment (P) via style transfer or CycleGAN, pseudo-label re-training (PL), and the proposed center-aware alignment (CA) that considers pixel-wise objectness and centerness. * indicates that pixel-level alignment is only applied during adapting from Sim10k to Cityscapes.

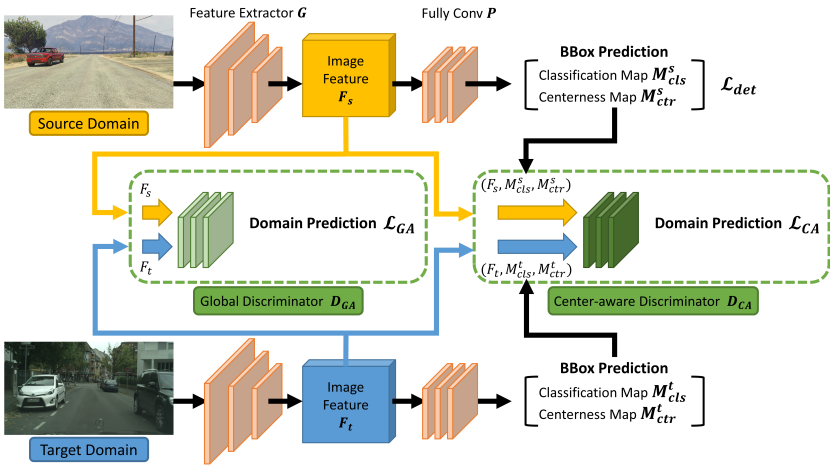
Method	G	I	L	P	PL	CA
DAF [2] CVPR'18	√	√				
SC-DA [41] CVPR'19	√	√				
SW-DA [32] CVPR'19	√		√	√*		
DAM [22] CVPR'19	√			√		
MAF [14] ICCV'19	√	√				
MTOR [1] CVPR'19	√	√				
STABR [21] ICCV'19	√				√	
PDA [15] WACV'20	√			√		
Ours	√					√

aligning lower-level features is more effective since global feature alignment suffers from the cross-domain variations of foreground objects and background clutter. To improve instance-level alignment, Zhu *et al.* [41] apply k-means clustering to group proposals and obtain the centroids of these clusters, which achieves a balance between global and instance-level alignment. However, their method introduces additional data-independent hyper-parameters for clustering and is not end-to-end trainable. Other variants improve feature alignment based on a hierarchical module [14], a style-transfer based method to address the source-biased issue [22], a teacher-student scheme to explore object relations [1], and a progressive alignment scheme [15].

While the above methods are based on two-stage detectors, Kim *et al.* [21] propose a one-stage adaptive detector for faster inference, via a hard negative mining technique for seeking more reliable pseudo-labels. However, their method only partially alleviates the issues brought by background and does not consider every pixel during feature alignment to reduce the domain gap. We also note that all aforementioned methods are based on anchors, in which performing instance-level alignment would be sensitive to inaccurate proposals in the target domain and the mixture of foreground/background pixels in a proposal. In contrast, we address these drawbacks by predicting pixel-wise objectness and proposing center-aware feature alignment, which only focuses on the discriminative parts of objects at the pixel scale. In Table 1, we summarize the alignment methods used in the aforementioned techniques for domain adaptive object detection.

3 Proposed Method

In this section, we first describe global feature alignment, and then introduce the proposed center-aware alignment that utilizes pixel-wise objectness and ceter-



4

Fig. 2. Proposed framework for domain adaptive object detection. Given the source and target images, we feed them to a shared feature extractor G to obtain their features F. Then, the global alignment on these features is performed via a global discriminator D_{GA} and a domain prediction loss $\mathcal{L}_{GA}$. Next, we pass the feature through the fully-convolutional module P to produce the classification and centerness maps. These maps and the feature F are utilized to generate the center-aware features. Finally, we use a center-aware discriminator D_{CA} and another domain prediction loss $\mathcal{L}_{CA}$ to perform the proposed center-aware feature alignment. Note that the bounding box prediction loss $\mathcal{L}_{det}$ is only operated on source images using their corresponding ground-truth bounding boxes.

ness. To improve the performance, we further incorporate multi-scale alignment that takes object scale into account during adaptation.

3.1 Algorithm Overview

Given a set of source images I_s, their ground-truth bounding boxes B_s, and unlabeled target images I_t, our goal is to predict bounding boxes B_t on the target image. To this end, we propose to utilize two alignment schemes that complement each other: global alignment that accounts for image-level distributions and the proposed center-aware alignment that focuses more on foreground pixels. The overall procedure is illustrated in Fig. 2. Given a shared feature extractor G across domains, we first extract features $F = G(I)$ and perform global alignment via using a global discriminator and a domain prediction loss. Second, followed by G, a fully-convolutional module P is adopted to predict pixel-wise objectness and centerness maps. Through combining these maps with the feature F, we employ another center-aware discriminator and its domain prediction loss to perform center-aware alignment.

3.2 Global Feature Alignment

The goal of global alignment is to align the feature maps on the image level to reduce the domain gap. To this end, we apply the adversarial alignment technique [2] via utilizing a global discriminator D_{GA}, which aims to identify whether the pixels on each feature map come from the source or the target domain.

Particularly, given the K-dimensional feature map $F \in \mathbb{R}^{H \times W \times K}$ of the spatial resolution $H \times W$ from the feature extractor G, the output of D_{GA} is a domain classification map that has the same size as F, while each location represents the domain label corresponding to the same location on F. Note that we set the domain label z of source and target domain as 1 and 0, respectively. Therefore, the discriminator can be optimized by minimizing the binary cross-entropy loss. For a location (u, v) on F, the loss function can be written as

$$\mathcal{L}_{GA}(I_s, I_t) = -\sum_{u,v} z \, \log(D_{GA}(F_s)^{(u,v)}) + (1-z)\log(1 - D_{GA}(F_t)^{(u,v)}). \quad (1)$$

To perform adversarial alignment, we apply the gradient reversal layer (GRL) [7] to feature maps of both source/target images, in which the sign of the gradient is reversed when optimizing the feature extractor via the GRL layer. Then the mechanism works as follows. The loss for the discriminator is minimized via (1), while the feature extractor is optimized by maximizing this loss, in order to deceive the discriminator. We also note that most existing methods (those in Table 1) utilize such global alignment that focuses on image-level distributions (i.e., more background pixels in reality). We also use global alignment in our framework to complement the proposed center-aware alignment that focuses on foreground pixels.

3.3 Center-aware Alignment

As mentioned in Section 1 and Table 1, existing methods [2,14,41] for instance-level alignment are based on proposals, and thus these approaches may suffer from the background effect. In order to address this issue, we propose a center-aware alignment method that allows us to focus on discriminative object regions. To this end, we adopt a center-aware discriminator D_{CA} for aligning features in the high-confidence area on the pixel level.

Definition. With a designed fully-convolutional network P (as detailed in Section 3.5) and feature map $F \in \mathbb{R}^{H \times W \times K}$ from the feature extractor G, we pass F through P, and obtain a classification output $M_{cls} \in \mathbb{R}^{H \times W \times C}$ and a class-agnostic centerness output $M_{ctr} \in \mathbb{R}^{H \times W}$, where C is the number of categories. Each location on the classification and centerness maps indicates corresponding objectness and centerness scores, respectively.

Discover Object Region. In order to find the confident area containing foreground objects, we utilize two cues derived from our object detector as mentioned above: 1) a class-agnostic map of the objectness scores and 2) a centerness map

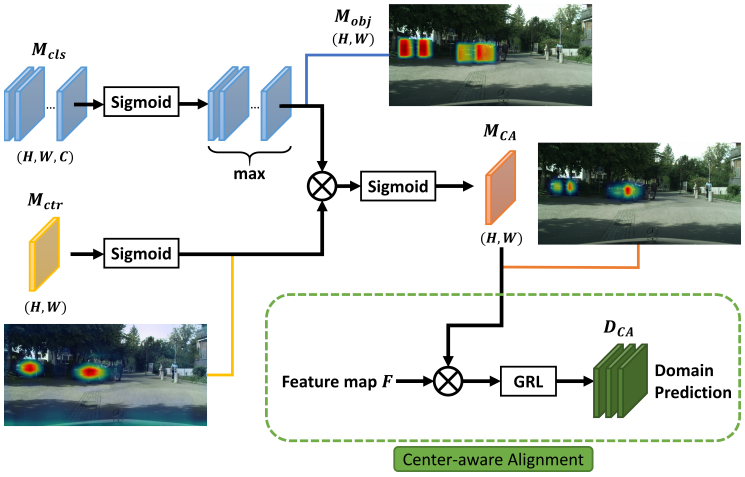


Fig. 3. Proposed center-aware alignment. Given the classification output M_{cls}, we first convert it to a class-agnostic map M_{obj}, which is then merged with the centerness output M_{ctr} into a center-aware map M_{CA} via (2) to identify potential object locations. Next, we use this map M_{CA} as the guidance to weight the global feature map F. Finally, this weighted feature map serves as the input to the center-aware discriminator D_{CA} to enable the proposed center-aware alignment in the feature space via (3).

that highlight object centers, so that the alignment can focus more on object parts. First, the objectness map can be obtained from the classification output M_{cls}. To obtain the class-agnostic map, we apply the *sigmoid* activation on each channel and take the *max* operation over categories. Similarly, the final class-agnostic centerness map is obtained via applying the *sigmoid* activation on the centerness output M_{ctr}. Overall, the final map M_{CA} to guide our center-aware alignment is calculated as follows:

$$\begin{aligned} M_{obj} &= \max_{c}(\sigma(M_{cls})), \\ M_{CA} &= \sigma(\delta\, M_{obj} \odot \sigma(M_{ctr})), \end{aligned} \tag{2}$$

where σ represents the *sigmoid* activation and $\odot$ denotes the element-wise product, i.e., Hadamard product, on the spatial maps. Since the values in M_{obj} and $\sigma(M_{ctr})$ are ranged from 0 to 1, a scaling factor δ is introduced for preventing the value from being too small after the multiplication. The factor δ is set to 20 in all experiments.

Perform Alignment. With the center-aware map M_{CA}, we are able to highlight the area where alignment on the pixel level should pay attention. To use this map as the guidance to our center-aware alignment, we multiply it by the feature

map F and then feed it into the center-aware discriminator D_{CA}:

$$\mathcal{L}_{CA}(I_s, I_t) = -\sum_{u,v} z \log(D_{CA}(M^s_{CA} \odot F_s)^{(u,v)}) + (1-z)\log(1 - D_{CA}(M^t_{CA} \odot F_t)^{(u,v)}). \quad (3)$$

We note that, since M_{CA} is a map of resolution $H \times W$, we duplicate it for K channels to compute its element-wise product with the feature map $F \in \mathbb{R}^{H \times W \times K}$. Then, we adopt a similar alignment process as described in (1) via the GRL layer. As a result, different from the global alignment method as described in Section 3.2, our model aligns pixel-wise features that are likely to be the object and hence mitigates the non-matching issue between foregrounds and backgrounds. The entire process of center-aware alignment is illustrated in Fig. 3.

3.4 Overall Objective for Proposed Framework

Given source images I_s, target image I_t, and the ground-truth bounding boxes B_s in the source domain, our goal is to predict bounding boxes B_t on the unlabeled target data. We have described the objective for feature alignment on both source and target images. Here, we introduce the details of the object detection objective on the source domain using I_s and B_s.

Objective for Object Detector. Motivated by the anchor-free detector [34], our fully-convolutional module P consists of the classification, centerness, and regression branches. The three branches output the objectness map M_{obj}, centerness map M_{ctr}, and regression map M_{reg}, respectively. For the classification and regression branches, their goals are to predict the classification score and the distance to the four sides of the corresponding object box for each pixel, respectively. We denote their loss functions as $\mathcal{L}_{cls}$ and $\mathcal{L}_{reg}$, which can be optimized via the focal loss [25] and IoU loss [39], respectively. For the centerness branch, it predicts the distance between each pixel and the center of the corresponding object box and can be optimized by the binary cross-entropy loss [34] denoted as $\mathcal{L}_{ctr}$. The overall objective for the detector on the source domain is:

$$\mathcal{L}_{det}(I_s, B_s) = \mathcal{L}_{cls} + \mathcal{L}_{reg} + \mathcal{L}_{ctr}. \quad (4)$$

Here, we omit the argument (I_s, B_s) of each loss function for simplicity.

Overall Objective. In order to obtain domain-invariant features across the source and target domains, we apply adversarial learning to feature maps using two discriminators, D_{GA} and D_{CA}, which perform the global alignment and center-aware alignment by minimizing the objective functions $\mathcal{L}_{GA}$ and $\mathcal{L}_{CA}$, respectively. The details can be found in Section 3.2 and Section 3.3. The overall loss function can be expressed as:

$$\mathcal{L}(I_s, I_t, B_s) = \mathcal{L}_{det}(I_s, B_s) + \alpha\mathcal{L}_{GA}(I_s, I_t) + \beta\mathcal{L}_{CA}(I_s, I_t), \quad (5)$$

where α and β are the weights used to balance the three terms.

3.5 Network Architecture and Discussions

Different from the prior work [2,14,41] that focuses on instance-level alignment, our center-aware feature alignment requires pixel-wise predictions for objectness and centerness maps, so we cannot directly adopt the network architecture in previous methods. In this section, we introduce our architecture via using a fully-convolutional module for producing pixel-wise predictions, as well as a multi-scale extension to account for the object scale during adaptation.

Network Architecture. As mentioned in Section 3.4, we connect feature map F with the fully-convolutional detection head P that contains three branches: the classification, centerness, and regression branches. Different from previous methods, all branches are constructed by the fully-convolutional network, so that the predictions are performed on the pixel level. Specifically, the three branches consist of four 3×3 convolutional layers, and each of them has 256 filters. For both discriminators in global and center-aware alignments, i.e., D_{GA} and D_{CA}, we use the same fully-convolutional architecture as the detection branch, in order to maintain the consistency of the output size and thus map to the original input image.

Multi-scale Alignment. We observe that such a fully-convolutional architecture is not robust to the object scale, which is crucial to the performance of feature alignment. Therefore, in the feature extractor G, we use the feature pyramid network (FPN) [24] to handle different sizes of objects. Particularly, FPN utilizes five levels of feature map, which can be denoted as F^i for $i = \{3, 4, ..., 7\}$. The feature map F^3 is responsible for the smallest objects, while the feature map F^7 focuses on the largest objects. Each of the feature maps in the pyramid, i.e., F^i, has 256 channels.

We connect each layer with one head that contains three detection branches and two discriminators, i.e., D_{GA} and D_{CA}, and thus the loss function in (5) can be extended to the feature map of each layer. As a result, we are able to align each individual feature map F^i via global and center-aware alignments via (1) and (3). It follows that each aligned layer is responsible for a certain range of object size while making the overall alignment process consistent.

How Pixel-wise Prediction Helps Feature Alignment. It is worth mentioning that we take advantage of the pixel-wise prediction for the following reasons: 1) Pixel-wise prediction does not involve any fixed anchor-related hyperparameters to produce proposals, which could be biased to the source domain during training; 2) Pixel-wise prediction considers all the pixels during training, which helps increase the capability of the model to identify the discriminative area of target objects; 3) The alignment can be performed on the pixel level and focuses on foreground pixels, which enables the model to learn better feature alignment. Note that the proposed method only depends on pixel-wise prediction, in which our method can be also applied to other similar detection models using the fully-convolutional module.

4 Experimental Results

We first provide the implementation details, and then describe datasets and evaluation metrics. Next, we compare our method with the state-of-the-art methods on multiple benchmarks. Finally, we conduct further analysis to understand the effect of each component in our framework. All the source code and models will be made available to the public.

4.1 Implementation Details

We implement our method with the PyTorch framework. In all the experiments, we set α and β in (5) as 0.01 and 0.1, respectively. Considering that center-aware alignment involves the detection output from (2), we first pre-train the detector only with the global alignment as a warm-up stage to ensure the reliability of detection before applying center-aware alignment and training the full objective in (5). Note that we set a larger α as 0.1 during pre-training for a faster convergence. For the adversarial loss using reversed gradients via GRL, we set the weight as 0.01 and 0.02 for D_{GA} and D_{CA}, respectively. The model is trained with learning rate of 5×10^{-3}, momentum of 0.9, and weight decay of 5×10^{-4}. The input images are resized with their shorter side as 800 and longer side less or equal to 1333.

4.2 Datasets

We follow the dataset setting as described in [2] and perform experiments for weather, synthetic-to-real and cross-camera adaptations on road-scene images.

Weather Adaptation. Cityscapes [3] is a scene dataset for driving scenarios, which are collected in dry weather. It consists of 2975 and 500 images in the training and validation set, respectively. The segmentation mask is provided for each image, consisting of eight categories: *person*, *rider*, *car*, *truck*, *bus*, *train*, *motorcycle* and *bicycle*. The Foggy Cityscapes [33] dataset is synthesized from Cityscapes as foggy weather. In the experiment, we adapt the model from Cityscapes to Foggy Cityscapes for studying the domain shift caused by the weather condition.

Synthetic-to-real. Sim10k [17] is a collection of synthesized images, which consists of 10,000 images and their corresponding bounding box annotations. We use images of Sim10k as the source domain, while Cityscapes is considered as the target domain. The adaptation from Sim10k to Cityscapes is used to evaluate the adaptation ability from synthesized to real-world images. Following the literature, only the class *car* is considered.

Cross-camera Adaptation. KITTI [8] is similar to Cityscapes as a scene dataset, except that KITTI has a different camera setup. The training set of KITTI consists of 7,481 images. We use the KITTI and Cityscapes as the source domain and target domain respectively, and evaluation the capability of cross-camera adaptation. Following the literature, only the class *car* is considered.

Table 2. Results of adapting Cityscapes to Foggy Cityscapes. The first and second groups adopt VGG-16 and ResNet-101 as the backbone, respectively. Note that results of each class are evaluated in $mAP^r_{0.5}$.

		Cityscapes → Foggy Cityscapes								
Method	Backbone	person	rider	car	truck	bus	train	mbike	bicycle	$mAP^r_{0.5}$
Baseline (F-RCNN)	VGG-16	17.8	23.6	27.1	11.9	23.8	9.1	14.4	22.8	18.8
DAF [2] CVPR'18		25.0	31.0	40.5	22.1	35.3	20.2	20.0	27.1	27.6
SC-DA [41] CVPR'19		33.5	38.0	48.5	26.5	39.0	23.3	28.0	33.6	33.8
MAF [14] ICCV'19		28.2	39.5	43.9	23.8	39.9	33.3	**29.2**	33.9	34.0
SW-DA [32] CVPR'19		29.9	**42.3**	43.5	24.5	36.2	32.6	30.0	35.3	34.3
DAM [22] CVPR'19		30.8	40.5	44.3	**27.2**	38.4	**34.5**	28.4	32.2	34.6
Ours (w/o adapt.)		30.5	23.9	34.2	5.8	11.1	5.1	10.6	26.1	18.4
Ours (GA)		38.7	36.1	53.1	21.9	35.4	25.7	20.6	33.9	33.2
Ours (CA)		41.3	38.2	56.5	21.1	33.4	26.9	23.8	32.6	34.2
Ours (GA+CA)		**41.9**	38.7	**56.7**	22.6	**41.5**	26.8	24.6	**35.5**	**36.0**
Oracle		47.4	40.8	66.8	27.2	48.2	32.4	31.2	38.3	41.5
Ours (w/o adapt.)	ResNet-101	33.8	34.8	39.6	18.6	27.9	6.3	18.2	25.5	25.6
Ours (GA)		39.4	41.1	54.6	23.8	42.5	31.2	25.1	35.1	36.6
Ours (CA)		40.4	**44.9**	**57.9**	24.6	**49.6**	32.1	25.2	34.3	38.6
Ours (GA+CA)		**41.5**	43.6	57.1	**29.4**	44.9	**39.7**	**29.0**	**36.1**	**40.2**
Oracle		44.7	43.9	64.7	31.5	48.8	44.0	31.0	36.7	43.2

4.3 Overall Performance

We compare our method with existing state-of-the-art approaches in Table 2 and Table 3, while the results evaluated by other metrics are provided in Table 4. We present two baselines: proposal-based Faster R-CNN [30] and our fully-convolutional detector denoted as "Ours (w/o adapt.)", both without adaptation. In all the tables, we denote global alignment and center-aware alignment as "GA" and "CA", respectively. To understand how much domain gap our model reduces, we also present the "Oracle" results, in which the model is trained and tested on the target domain using our model. Moreover, we consider two backbone architectures as our feature extractor: VGG-16 [19] or ResNet-101 [13].

Weather Adaptation. In Table 2, we notice that our baseline without adaptation performs similarly (i.e., around 18%) to the F-RCNN baseline, using the VGG-16 backbone. After adaptation, our method (GA + CA) improves our baseline by 17.6% and performs the best compared to other methods in $mAP^r_{0.5}$, especially against the ones [2,14,41] that adopt both global and instance-level alignments. Overall, for both architectures, we consistently show that using the proposed center-aware alignment performs better than global alignment, and combining both is complementary and achieves the best performance.

Synthetic-to-real. In the left part of Table 3, we show that our final model (GA+CA) using the VGG-16 backbone performs favorably against existing methods. We note that, compared to a recent method, SW-DA* [32], that adds the

Table 3. Results of adapting Sim10k/KITTI to Cityscapes. The first and second groups adopt VGG-16 and ResNet-101 as the backbone, respectively. The symbol $*$ indicates that additional training images generated via pixel-level adaptation are used.

		Sim10k	KITTI
Method	Backbone	$mAP^r_{0.5}$	$mAP^r_{0.5}$
Baseline (F-RCNN)	VGG-16	30.1	30.2
DAF [2] CVPR'18		39.0	38.5
MAF [14] ICCV'19		41.1	41.0
SW-DA [32] CVPR'19		42.3	-
SW-DA* [32] CVPR'19		47.7	-
SC-DA [41] CVPR'19		43.0	42.5
Ours (w/o adapt.)		39.8	34.4
Ours (GA)		45.9	39.1
Ours (CA)		46.6	41.9
Ours (GA+CA)		**49.0**	**43.2**
Oracle		69.7	69.7
Ours (w/o adapt.)	ResNet-101	41.8	35.3
Ours (GA)		50.6	42.3
Ours (CA)		51.1	43.6
Ours (GA+CA)		**51.2**	**45.0**
Oracle		70.4	70.4

Table 4. More mAP metrics of adapting Sim10k/KITTI to Cityscapes using ResNet-101 as the backbone.

	Sim10k → Cityscapes						KITTI → Cityscapes					
Method	mAP	$mAP^r_{0.5}$	$mAP^r_{0.75}$	mAP^r_S	mAP^r_M	mAP^r_L	mAP	$mAP^r_{0.5}$	$mAP^r_{0.75}$	mAP^r_S	mAP^r_M	mAP^r_L
Ours (w/o adapt.)	23.1	41.8	22.4	5.1	26.8	46.6	15.9	35.3	12.8	1.5	17.8	36.5
Ours (GA)	26.4	50.6	25.2	5.7	26.3	57.3	18.8	42.3	14.7	5.0	24.5	35.9
Ours (CA)	26.8	51.1	26.3	**7.5**	27.9	54.6	20.3	43.6	17.3	4.1	25.4	40.8
Ours (GA+CA)	**28.6**	**51.2**	**27.4**	7.1	**30.2**	**58.3**	**22.2**	**45.0**	**20.0**	**5.3**	**28.1**	**43.1**
Oracle (ResNet-101)	44.6	70.4	46.2	15.7	49.2	79.2	44.6	70.4	46.2	15.7	49.2	79.2

augmented data into training via the pixel-level adaptation technique, our result is still better than theirs. We also notice that the improvement from GA-only to GA+CA using the ResNet-101 backbone is not significant. However, we will show that more performance gain can be achieved when using other mAP metrics with a higher standard later.

Cross-camera Adaptation. In the right part of Table 3, we show that our method achieves favorable performance against others, and adding CA consistently improves the results, e.g., 8.8% and 9.7% gain compared to the baseline without adaptation, using VGG-16 or ResNet-101, respectively.

More Discussions. Although the CA-only model performs competitively against the GA-only model, they essentially focus on different tasks. For global alignment, it tries to align image-level distributions, which is necessary to help reduce

the domain gap but may focus too much on background pixels. For our center-aware alignment, we focus more on pixels that are likely to be the foreground, in which the alignment process considers foreground distributions more. As such, they act as a different role, in which combing both is complementary to further improve the performance (i.e., GA+CA).

In addition, in Table 2, we notice that the performance of some categories that are underrepresented such as *truck* and *mbike* is lower than that of other categories. One reason is that these categories contain less foreground pixels in the source domain, in which our center-aware alignment may pay less attention to them. One could adopt a stronger backbone (e.g., ResNet-101 in Table 2) to improve the performance or use the category prior that allows the model to focus more on those underrepresented categories, which is not in the scope of this work and could be one future work.

4.4 More Results and Analysis

In this section, we provide detailed analysis in the proposed method with more mAP measurements. In addition, we visualize our center-aware maps and more results are provided in the supplementary material.

More mAP Metrics. In Table 4, we show more mAP metrics than $\text{mAP}^r_{0.5}$, to analyze where our method helps the detector adapting to different scenarios. On the Sim10k case, as discussed in Section 4.3, we observe that our full model using ResNet-101 does not improve $\text{mAP}^r_{0.5}$ a lot compared with the GA-only model. However, we show that under a more challenging case, e.g., $\text{mAP}^r_{0.75}$, mAP^r_S and mAP^r_M, adding CA improves results over GA-only by 2.2%, 1.4%, and 3.9%, respectively. It validates the usefulness of our center-aware alignment for challenging adaptation cases. Similar observations could be found in the KITTI case. Such measurements also suggest an interesting aspect for domain adaptive object detection to better understand its challenges.

Multi-scale Alignment. To verify the effectiveness of our multi-scale alignment scheme, we conduct an ablation study on Sim10k $\rightarrow$ Cityscapes using the ResNet-101. In Table 5, we compare results using all the scales ($F^3 \sim F^7$), three scales ($F^5 \sim F^7$) via removing the bottom two scales, three scales ($F^3 \sim F^5$) via removing the top two scales, and a single scale F^5. Note that, we choose the single scale as F^5 since it is the middle scale, which has the most influential impact. We show that adding more scales gradually improves the performance on all the metrics, which validates the usefulness of our proposed multi-scale alignment method. Moreover, $F^3 \sim F^5$ is responsible for smaller objects, in which the $\text{mAP}^r_S/\text{mAP}^r_M$ results are better than the $F^5 \sim F^7$ ones. In contrast, mAP^r_L is better for $F^5 \sim F^7$ as it handles larger objects. This indicate that our multi-scale alignment is effective for handling various size of objects.

Qualitative Analysis. We first show some example results of the response map that our method tries to localize the object. In Fig. 4, the baseline without adaptation has difficulty to find any object centers, while our global alignment method

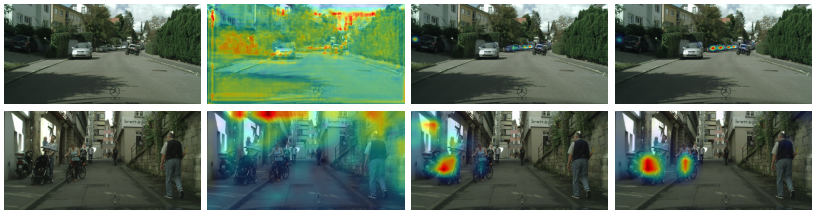


Fig. 4. Comparisons of response maps on Sim10k-to-Cityscapes. The maps on the first row are extracted from the feature layer F^3 which focuses on smaller objects, while the second row is for the feature layer F^6. After adding the proposed center-aware alignment, the model could focus more on the objects and reduce background noises.

Table 5. Ablation study of our multi-scale alignment using ResNet-101.

Sim10k → Cityscapes						
Aligned Scale	mAP	$mAP^r_{0.5}$	$mAP^r_{0.75}$	mAP^r_S	mAP^r_M	mAP^r_L
w/o adapt.	23.1	41.8	22.4	5.1	26.8	46.6
F^5	24.2	48.9	22.4	5.7	24.0	52.4
$F^3 \sim F^5$	26.2	48.7	25.0	6.9	28.7	53.0
$F^5 \sim F^7$	26.1	49.2	25.8	6.2	26.8	54.8
$F^3 \sim F^7$	**28.6**	**51.2**	**27.4**	**7.1**	**30.2**	**58.3**

is able to localize some objects. Adding the proposed center-aware alignment enables our method to discover more object centers at different object scales. We also note that, each scale in our model may focus on a different size of object, e.g., the upper example in Fig. 4 may miss larger objects. However, those objects missing at a smaller scale could be identified at another scale.

5 Conclusions

In this paper, we propose a center-aware feature alignment method to tackle the task of domain adaptive object detection. Specifically, we propose to generates pixel-wise maps for localizing object regions, and then use them as the guidance for feature alignment. To this end, we develop a method to discover center-aware regions and perform the alignment procedure via adversarial learning that allows the discriminator to focus on features coming from the object region. In addition, we design the multi-scale feature alignment scheme to handle different object sizes. Finally, we show that incorporating global and center-aware alignments improves domain adaptation for object detection and achieves state-of-the-art performance on numerous benchmark datasets and settings.

Acknowledgment. This work was supported in part by the Ministry of Science and Technology (MOST) under grants MOST 107-2628-E-009-007-MY3, MOST 109-2634-F-007-013, and MOST 109-2221-E-009-113-MY3, and by Qualcomm through a Taiwan University Research Collaboration Project. M.-H. Yang is supported in part by NSF CAREER Grant 1149783.

References

1. Cai, Q., Pan, Y., Ngo, C.W., Tian, X., Duan, L., Yao, T.: Exploring object relation in mean teacher for cross-domain detection. In: CVPR (2019) 1, 4
2. Chen, Y., Li, W., Sakaridis, C., Dai, D., Gool, L.V.: Domain adaptive faster r-cnn for object detection in the wild. In: CVPR (2018) 1, 2, 3, 4, 6, 9, 10, 11, 12
3. 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: CVPR (2016) 3, 10
4. Dai, J., He, K., Sun, J.: Instance-aware semantic segmentation via multi-task network cascades. In: CVPR (2016) 1
5. Dai, S., Sohn, K., Tsai, Y.H., Carin, L., Chandraker, M.: Adaptation across extreme variations using unlabeled domain bridges. arXiv preprint arXiv:1906.02238 (2019) 1
6. Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., Tian, Q.: Centernet: Keypoint triplets for object detection. In: ICCV (2019) 3
7. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: ICML (2015) 1, 6
8. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: CVPR (2012) 3, 10
9. Girshick, R.: Fast r-cnn. In: ICCV (2015) 1
10. Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Simultaneous detection and segmentation. In: ECCV (2014) 1
11. Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Hypercolumns for object segmentation and fine-grained localization. In: CVPR (2015) 1
12. He, K., Gkioxari, G., Dollar, P., Girshick, R.: Mask R-CNN. In: ICCV (2017) 1
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016) 11
14. He, Z., Zhang, L.: Multi-adversarial faster-rcnn for unrestricted object detection. In: ICCV (2019) 1, 2, 4, 6, 9, 11, 12
15. Hsu, H.K., Yao, C.H., Tsai, Y.H., Hung, W.C., Tseng, H.Y., Singh, M., Yang, M.H.: Progressive domain adaptation for object detection. In: WACV (2020) 1, 4
16. Inoue, N., Furuta, R., Yamasaki, T., Aizawa, K.: Cross-domain weakly-supervised object detection through progressive domain adaptation. In: CVPR (2018) 1
17. Johnson-Roberson, M., Barto, C., Mehta, R., Sridhar, S.N., Rosaen, K., Vasudevan, R.: Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? In: ICRA (2017) 3, 10
18. Kang, K., Li, H., Yan, J., Zeng, X., Yang, B., Xiao, T., Zhang, C., Wang, Z., Wang, R., Wang, X., less, W.O.: T-cnn: Tubelets with convolutional neural networks for object detection from videos. TCSVT (2018) 1
19. Karen, S., Andrew, Z.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015) 11
20. Karpathy, A., Li, F.F.: Deep visual-semantic alignments for generating image descriptions. In: CVPR (2015) 1
21. Kim, S., Choi, J., Kim, T., Kim, C.: Self-training and adversarial background regularization for unsupervised domain adaptive one-stage object detection. In: ICCV (2019) 1, 4
22. Kim, T., Jeong, M., Kim, S., Choi, S., Kim, C.: Diversify and match: A domain adaptive representation learning paradigm for object detection. In: CVPR (2019) 1, 4, 11

23. Law, H., Deng, J.: Cornernet: Detecting objects as paired keypoints. In: ECCV (2018) 3
24. Lin, T.Y., Dollar, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: CVPR (2017) 9
25. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollar, P.: Focal loss for dense object detection. In: ICCV (2017) 3, 8
26. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: ECCV (2016) 1, 3
27. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR (2015) 3
28. Long, M., Cao, Y., Wang, J., Jordan, M.I.: Learning transferable features with deep adaptation networks. In: ICML (2015) 1
29. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified and real-time object detection. In: CVPR (2016) 1
30. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: NeurIPS (2015) 3, 11
31. Saito, K., Watanabe, K., Ushiku, Y., Harada, T.: Maximum classifier discrepancy for unsupervised domain adaptation. In: CVPR (2018) 1
32. Saito1, K., Ushiku, Y., Harada, T., Saenko, K.: Strong-weak distribution alignment for adaptive object detection. In: CVPR (2019) 1, 3, 4, 11, 12
33. Sakaridis, C., Dai, D., Gool, L.V.: Semantic foggy scene understanding with synthetic data. IJCV (2018) 3, 10
34. Tian, Z., Shen, C., Chen, H., He, T.: Fcos: Fully convolutional one-stage object detection. In: ICCV (2019) 3, 8
35. Tsai, Y.H., Sohn, K., Schulter, S., Chandraker, M.: Domain adaptation for structured output via discriminative patch representations. In: ICCV (2019) 1
36. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: CVPR (2017) 1
37. Wu, Q., Shen, C., Wang, P., Dick, A., van den Hengel, A.: Image captioning and visual question answering based on attributes and external knowledge. TPAMI (2018) 1
38. Xu, K., Ba, J.L., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R.S., Bengio, Y.: Show and attend and tell: Neural image caption generation with visual attention. In: ICML (2015) 1
39. Yu, J., Jiang, Y., Wang, Z., Cao, Z., Huang, T.: Unitbox: An advanced object detection network. In: ACMMM (2016) 8
40. Zhou, X., Zhuo, J., Krähenbühl, P.: Bottom-up object detection by grouping extreme and center points. In: CVPR (2019) 3
41. Zhu, X., Pang, J., Yang, C., Shi, J., Lin, D.: Adapting object detectors via selective cross-domain alignment. In: CVPR (2019) 2, 4, 6, 9, 11, 12