

Multi-Scale Adversarial Cross-Domain Detection with Robust Discriminative Learning

YoungSun Pan¹, Andy J Ma^{1,3*}, Yuan Gao¹, JinPeng Wang², and Yiqi Lin¹

¹School of Data and Computer Science, Sun Yat-sen University, China.

²School of Electronics and Information Technology, Sun Yat-sen University, China.

³Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, China

Abstract

Domain shift practically exists in almost all computer vision tasks including object detection, caused by which the performance drops evidently. Most existing methods for domain adaptation are specially designed for classification. For object detection, existing methods separate domain shift into image-level shift and instance-level shift and align image-level feature and instance-level feature respectively. However, we find that there are two problems which remain unsolved yet. First, the scale of objects is not the same even in an image. Second, negative transfer can affect model performance if not handled properly. We improve the performance of cross-domain detection from three perspectives: 1) using multiple dilated convolution kernels with different dilation rate to reduce the image-level domain discrepancy; 2) removing images or instances with low transferability to weaken the influence of negative transfer; 3) diversifying distributions by keeping instances' feature away from each other, and then pull them closer to the center of each category, so that make source samples distribution more dispersed and more robust for cross-domain detection. We test our model with Cityscapes [5], Foggy Cityscape [30] and SIM 10K [18] datasets, experimental results show that our method outperforms the state-of-the-art for object detection under the setting of unsupervised domain adaptation (UDA).

1. Introduction

Object detection aims to identify and localize all interesting object instances in images. Currently, many deep convolutional networks (CNNs) based methods [27, 11, 10, 28, 16] have been proposed which achieve impressive perfor-



(a) Foggy Cityscapes. (b) Cityscapes.



(c) SIM 10k.

Figure 1. Illustration of some samples of different datasets: although all of them are urban images, the style is very different, caused by weather, illumination, acquisition sensor and so on, which is a problem for ordinary object detection method.

mance. This achievement is built on large amount of dense annotations obtained by expensive human labor [7, 22]. While in realistic practice, the performance drops drastically caused by the large variance such as backgrounds, illumination, viewpoints, *etc.* (see in Figure 1). The reason lies in the different data distributions of domains, typically known as domain shift [14]. The typical solution is to finetune the trained deep models on task-specific datasets which may be prohibitively expensive to collect enough labeled data. To address this issue, unsupervised domain adaptation (UDA) methods [8, 34, 6] transfer discriminative features from related labeled source domains to unlabeled target data without extra annotations.

*Corresponding author. Email: majh8@mail.sysu.edu.cn.

This paper dedicates to cross-domain object detection problem. The existing domain adaptation methods for object detection [4, 36, 29] have achieved excellent performances, they mainly focus on aligning image-level and instance-level representations of source and target domains. In this paper, we improve Faster R-CNN on target datasets from three novel perspectives.

Firstly, in order to align image-level feature from different scales, we utilize multi-channel void convolution to extract multi-scale features. And then a 1×1 convolution is used for channel communication and outputting domain prediction. Secondly, for avoiding negative transfer, we discard the feature with low transferability on image-level and instance-level. Low transferability denoted as that samples easy to distinguished by domain discriminator. More specifically, target samples whose output of domain discriminators lower than a threshold are rejected. Thirdly, we extend metric learning to cross-domain object detection by diversifying features in a batch to restrain overfitting and centering features of the same category to preserve discrimination.

In a word, our contributions are summarized as follows:

- **Multi-Scale Adaptation.** We adopt multi-channel void convolutions with different void rates at image-level adaptation to align source and target domains in different scales.
- **Reduce Negative Transfer.** During the training process, we reduce negative transfer by rejecting low transferability target data which decided by domain discriminators.
- **Robust Discriminative distribution.** A novel method is proposed to diversify domain distribution and center category distribution for robust discriminative distribution. The extended experiments indicate that this composition can improve cross-domain robustness for object detection task.

2. Related Work

2.1. Object Detection

With the rise of CNNs, object detection methods have made remarkable progress. R-CNN [11] is the first model that trains a network to classify all regions of interest (ROI) extracted by selective search from images. Fast R-CNN [10] increases speed by sharing the feature map of all ROIs, and presented ROI pooling which map features of different sizes to the same size to solve the problem of the different sizes of ROIs. Faster R-CNN [28] firstly utilizes Region Proposal Network (RPN) to extract ROIs, which is much better than selective search used by [11, 10] in terms of speed and accuracy. It achieved state-of-the-art performance and followed by many works [9, 16, 4, 20]. However, all of those models did not consider the scenario of

cross-domain detection. Our method in this paper is based on Faster-RCNN and further reduce domains' discrepancy.

2.2. Domain Adaptation

For classification tasks, domain adaptation has been widely researched. There are many methods proposed to narrow the gap between different domains, including narrowing Maximum Mean Discrepancy (MMD) [35, 24, 25], covariance matrix alignment [32], subspace alignment [14], geodesic flow kernel [12, 14], *etc.* Generative Adversarial Networks (GANs) [13] achieved great success in generating pictures by minimizing the JS divergence [13] or Wasserstein distance [1, 15] between two distributions. Domain Adversarial Neural Network (DANN) [8] and Adversarial Discriminative Domain Adaptation (ADDA) [34] used GANs to align different domains by training a domain classifier to classify the feature maps from which domain, and enforcing feature extractor to confuse the domain classifier. Joint Adaptation Network (JAN) [25] uses both adversarial learning and multiple kernel learning.

Different from those works, our model beyond classification problem and focus on object detection task.

2.3. Cross-domain Object Detection

While domain adaptation for classification has already received a lot of attention, only a few works [4, 36, 3, 29] consider domain adaptation for object detection. Domain Adaptive Faster R-CNN (DA Faster R-CNN) [4] train two domain classifier to align image-level features and instance-level features that from different domains, and enforce consistency between the outputs of two domain classifiers to improve the cross-domain robustness of RPN. Few-Shot Adaptive Faster R-CNN (FSA Faster R-CNN) [36] randomly select 9 bounding boxes with different scales and proportions of each image as inputs of image-level domain classifier. Mean Teacher for Cross-Domain Detection [3] firstly use mean teacher [33] in cross-domain detection for better cross-domain robustness. [29] mitigates negative transfer by utilizing weak alignment of high-level features that use focal loss [21] to reweight samples' weights that highlight hard-to-distinguish domain samples.

2.4. Unsupervised Metric Learning

Metric Learning is successfully applied in unsupervised learning and cross-domain person reidentification (ReID)[39, 38]. [38] train model with unlabeled data, by push feature vector from other vectors in the memory bank. The model classifies data with a special classifier named non-parametric softmax classifier that predicts the data with k (k is a super parametric) nearest neighbor. [39] is a method of cross-domain person reidentification. [39] diversify domain distribution by making every image identifiable, and [39] pulls its feature close with the k nearest

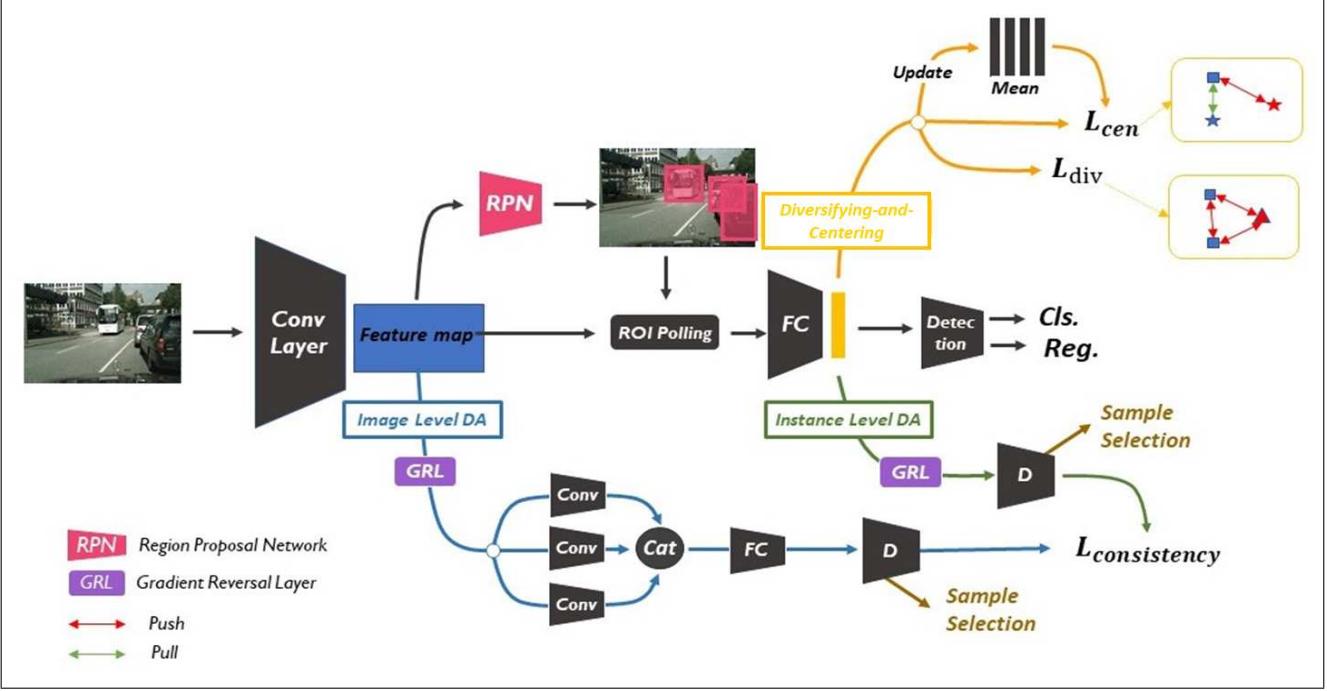


Figure 2. Overview of the proposed model: we narrow the domain shift with image-level and instance-level adversarial domain adaptation modules. While instance-level module tackles multi-scale with ROI pooling, image-level address it with multi-channel void convolutions. To reduce negative transfer, we reject target samples with low transferability. We further present diversifying-and-centering module which diversify domain distribution and center feature vectors of each class (stars in the figure). More details can be obtained in section 4.

neighbor. But those models require a memory bank to storage all feature vectors of the dataset, which is unachievable in object detection because bounding boxes obtained by RPN are not fixed, and features of an instance are different for unfixed bounding boxes, that is why an instance isn't appropriate to stored as a feature vector. In this paper, we present a novel method to diversify domain distribution and center category distribution for cross-domain object detection without memory bank.

3. Preliminaries

3.1. Faster R-CNN

Faster R-CNN [28], a typical and successful two-stage detector, is the baseline model in our work. It mainly consists of three major part: 1) shared backbone convolutional network; 2) a region proposal network (RPN) that produce region-of-interest (ROIs); 3) an ROI based classifier. The architecture is shown in the upper left part of figure 2. At first, a backbone network generates a feature map with a single image as input. And then RPN produces bounding boxes of ROIs based on the feature map. At last, ROI based classifier predicts the category label of all feature vectors obtained from ROI-pooling which maps bounding boxes to feature vectors of the same size. The loss of Faster R-CNN

consists of two parts, RPN loss and ROI loss:

$$L_{det} = L_{rpn} + L_{ROI} \quad (1)$$

Both RPN loss and ROI loss consist of classification part and regression part, which respectively indicate how accurate the predicted category probability are and how precise bounding boxes are. More details can be revisited in the original paper [28].

3.2. Domain Adaptive Faster R-CNN

To the best of our knowledge, Domain Adaptive Faster R-CNN [4] which is followed by our method is the first work that specializes in unsupervised domain adaptation for object detection. It mainly including image-level adversarial adaptation and instance-level adversarial adaptation. What's more, to make RPN more robust for cross-domain detection, it enforces consistency between the above two domain classifiers of different levels. Therefore, it's loss can be written as:

$$L = L_{det} + \lambda(L_{img} + L_{ins} + L_{cst}) \quad (2)$$

where L_{det} is detection loss of source data as Equation (1), L_{img} and L_{ins} are adaptation loss of two levels described above, L_{cst} is consistency loss of those two adversarial domain discriminator. More details and derivation can be obtained in [4]. We represent domain label with y_i , that $y_i =$

0 indicates the i -th training image from the source domain and $y_i = 1$ indicates it from the target domain. So, L_{img} , L_{ins} , L_{cst} are written as:

$$L_{img} = - \sum_i [y_i \log D_{img}(f_i) + (1 - y_i) \log D_{img}(1 - f_i)] \quad (3)$$

$$L_{ins} = - \sum_{i,j} [y_i \log D_{ins}(f_{i,j}) + (1 - y_i) \log D_{ins}(1 - f_{i,j})] \quad (4)$$

$$L_{cst} = \sum_{i,j} \|D_{img}(f_i) - D_{ins}(f_{i,j})\|_2 \quad (5)$$

where D_{img} is image-level domain discriminator, and the i -th image feature map is denoted as f_i . We denote D_{ins} as instance-level domain classifier, $f_{i,j}$ as the j -th region proposal feature vector in the i -th image. And $\|\cdot\|_2$ is \mathcal{L}_2 distance.

4. Method

4.1. Problem Setup

Under the classic setting of unsupervised domain adaptation, labeled data (X_S, Y_S) of source domain \mathcal{S} , and unlabeled data (X_T) of target domain \mathcal{T} are available, where X_S and X_T are input images of source domain and target domain respectively, Y_S denotes bounding box and object categories annotation for X_S . And our task is to learn an object detection model to predict objects' bounding box and categories of target domain.

4.2. Multi-Scale Adaptation

In this section, we will introduce multi-scale image-level domain adaptation and instance domain adaptation. As shown in Figure 1, the scale of different objects varies greatly even in the same image, *e.g.* for picture (b) the car on the right part is much bigger than the person and distant cars. While instance-level adaptation can solve this problem through ROI pooling, it's meaningful to do something for overcoming the problem at image level.

Image-level feature refers to the feature map outputs of the shared backbone network, *i.e.* the blue box in the left part of Figure 2. As shown in the image-level DA part of Figure 2, we apply a patch-based domain classifier to align domain distributions. At first, the feature map of each image input in multiple dilated convolutions with different dilation rates that indicate multi scales. Then using a 1×1 convolution for channel communication and to predict where the feature comes from, source domain or target. So that multi-scale image-level adaptation loss is the same as equation 3.

We align the domain distributions through a min-max game: domain discriminator D_{img} , optimized by minimizing the above adversarial loss aims to distinguish which do-

main is the feature belongs to, meanwhile backbone network, optimized by maximizing the loss aims to confuse D_{img} . To play this min-max game, we utilize the gradient reverse layer (GRL) [8], which reverse signs while gradient pass through GRL during the back-propagation process.

Instance-level feature refers to the feature vectors obtained from ROI pooling (*i.e.* the orange rectangle in Figure 2). Owing to ROI pooling, instance-level features are independent of scale, because all instances' feature with different scales is mapped to the same size. So, for instance-level adaptation (equation 4) we follow DA Faster R-CNN [4], and use consistency (equation 5) to align RPN.

4.3. Sample Selection

Negative transfer is a problem of domain adaptation that cannot be ignored [26]. On this problem, [23, 37] reweight the adaptation loss that give samples hard to distinguish which category it belongs to low weight, samples easy to distinguish high weight, for that difficult to distinguish means low transferability. On the other side, [17, 2, 19] reject target samples if their transferability score low than a threshold. In this paper, we define a dynamic threshold γ , which is initially close to 0.5, and gradually down to a constant β ($0 < \beta < \gamma$).

Because with the training going on, the domain differences will become smaller, and we expect more data to participate in training. If a sample's (image-level feature or instance-level) transferability score is not greater than the threshold γ , only the corresponding domain discriminator will be optimized by the sample's adaptation loss instead of both backbone network and domain discriminator.

[23, 37] denote transferability score as entropy criterion $H(C) = - \sum_i^n C_n \log C_n$, where n is the number of classes and C_n is the probability of model predicting an image to class n . Low entropy means that the model is confident in predicting the image, so the image has a high transferability score. Under open set domain adaption (OSDA) setting that both domains contain unknown classes, [17, 2, 19] denote transferability score as the output of domain classifier and reject these easy-to-distinguish samples as unknown classes.

For Faster R-CNN model [28], it's not suitable to denote transferability as entropy. Because unlike the classification task, prediction of ROI based classifier is greatly affected by the bounding box. We use two domain classifiers' prediction to measure transferability of images and instances (see in figure 2).

4.4. Robust Discriminative Distribution

In this section, we will introduce Robust Discriminative Distribution of our model. This module contains two parts, diversifying operation and centering operation (DaC). As shown in the right upper part in Figure 2, the long orange

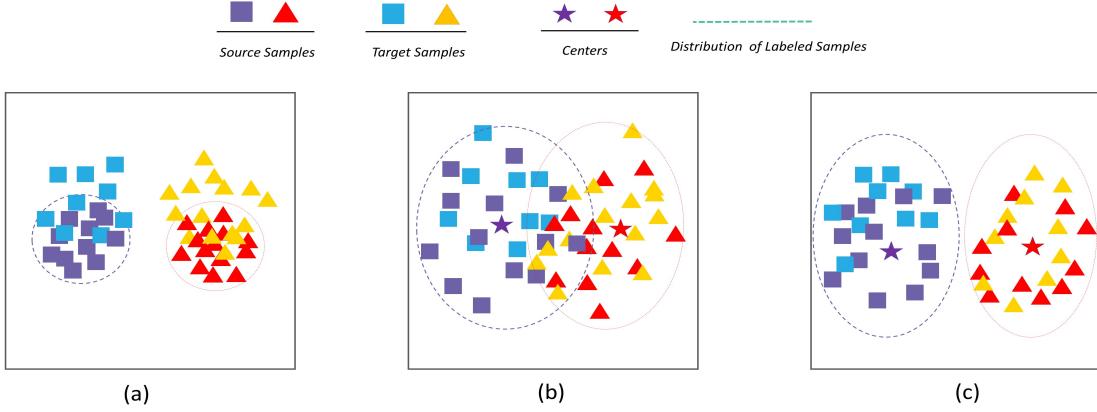


Figure 3. A comparison of General CNNs, diversifying and diversifying-and-centering. (a) General CNNs: samples of source domain may be dense (*i.e.* distribution is small) so is not robustness for domain shift. (b) Diversifying: we push samples away from each other to diversify domain distribution, however, it may cause category confusion. (c) Diversifying-and-Centering: combining diversifying with centering, the category confusion is weakened. Model (c) is outperforming model (a) in cross-domain robustness.

rectangle is a feature vector of an instance, and the square denotes the average value of each class. For diversifying operation (see the box named L_{div}), blue squares and red triangles denote as instances of an image, we measure their distance by cosine distance inspired by [38], and push them away from each other. The red two-way arrow means pushing away. Hence, diversifying loss can be written as:

$$L_{div} = \sum_i \sum_{j \neq i} \cos(O_i, O_j) \quad (6)$$

where O_i and O_j are the i -th and j -th instance's vector respectively of the image.

For centering operation (see the box named L_{cen}), the blue square is an instance's feature vector and the blue star is the average value of the class that the instance belongs to, red star denotes centers of other classes. While red two-way arrow means pushing away, green means pulling close. centering loss is written as:

$$L_{cen} = \sum_i \sum_{j \neq y_i} \cos(O_i, M_j) - \sum_i \cos(O_i, M_{y_i}) \quad (7)$$

where y_i is the category label of i -th instance, and M_j is the mean of j -th class vectors. However, we have no idea to use memory bank to store all objects' feature vector of dataset just like [39, 38]. Because one image's bounding boxes produced by RPN are not fixed, we can't storage all instances' feature vector. That is why directly computing average of every class is not easy, so we use the momentum method to estimate them. For each instance O_i in an image, we update averages by:

$$M_{y_i} = m(M_{y_i}) + (1 - m)O_i \quad (8)$$

where m is the momentum scalar between 0 and 1. Let's make m close to 1, so that M_{y_i} approximate to the average of y_i -th class. In order to better explain DaC we show the comparison of general CNN and DaC in Figure 3

4.5. Overview

Figure 2 shows an overview of our method. Our work is based on Faster R-CNN [28], and we improve it with several domain alignment components so that our model is applicable to cross-domain object detection task. Compared with Domain Adaptive Faster R-CNN model [4], our method takes into consideration the problem of multi-scale and negative transfer. The left upper of Figure 2 is the structure of Faster R-CNN. The multi-scale image-level domain module is added after the image-level feature map (blue parallelogram of Figure 2). The instance-level domain classifier is used to align the ROI-wise feature vectors (orange rectangle). And then a consistency component links above two domain classifiers to make RPN perform better in the target domain. The right upper part of Figure 2 is robust discriminative distribution component, which pushes feature vector away from each other and pulls them close to the class center. So that we can diversify the domain distribution as well as centering each class's distribution. At last, to narrow the influence of negative transfer, we reject those images and instances whose transferability score is lower than the threshold γ (sample selection in figure 2).

For labeled source images, we utilize original Faster R-CNN loss:

$$L_{sdet} = \mathbb{E}_{(x,y) \in \mathcal{S}} L_{det}(x, y) \quad (9)$$

As 4.3 mentioned, we divide target domain \mathcal{T} into two

	Mul	Rej	DaC	person	rider	car	truck	bus	train	mcycle	bicycle	mAP	
Faster R-CNN [28]				24.1	29.9	32.7	10.9	13.8	5.0	14.6	27.9	19.9	
ADDA [34]				25.7	35.8	38.5	12.6	25.2	9.1	21.5	30.8	24.9	
FSA Faster R-CNN [36]				29.1	39.7	42.9	20.8	37.4	24.1	26.5	29.9	31.3	
Strong-Weak DA [29]				29.9	42.3	43.5	24.5	36.2	32.6	30.0	35.0	34.3	
DA Faster R-CNN [4](baseline)				25.0	31.0	40.5	22.1	35.3	20.2	20.0	27.1	27.6	
ours		✓		27.4	39.7	41.6	20.7	35.4	27.1	22.1	32.2	30.8	
ours		✓	✓		30.3	42.4	44.8	18.8	41.3	39.6	28.6	33.8	35.1
ours		✓	✓	✓	32.3	44.0	46.8	20.8	43.3	45.8	29.7	33.4	37.0

Table 1. Quantitative results of our method and baselines on weather transfer scenario. Mul denotes multi-scale domain adaptation component, Sel means sample selection, DaC is diversifying-and-centering.

parts, \mathcal{T}^{sel} whose transferability score is higher than γ , and \mathcal{T}^{rej} that are rejected for low transferability score. So domain adaptation loss can be described as:

$$L_d^{sel} = \mathbb{E}_{(x \in \mathcal{S} \cup \mathcal{T}^{sel})} (L_{img}(x) + L_{ins}(x) + L_{cst}(x)) \quad (10)$$

$$L_d^{rej} = \mathbb{E}_{(x \in \mathcal{T}^{rej})} (L_{img}(x) + L_{ins}(x) + L_{cst}(x)) \quad (11)$$

where L_{img} , L_{ins} and L_{cst} can be obtained from equation 3, 4, 5 respectively.

For DaC described at 4.4, we do not use centering operation in the unlabeled target domain because bounding boxes and categories of target domain aren't labeled. So DaC loss is:

$$L_{DaC} = \mathbb{E}_{(x, y) \in \mathcal{S}} (L_{div}(x) + L_{cen}(x, y)) + \mathbb{E}_{x \in \mathcal{T}} L_{div}(x) \quad (12)$$

L_{div} and L_{cen} are shown in equation 6 and 7.

Over all, the final objective of our model is delivering the optimal $(\hat{\theta}_f, \hat{\theta}_c, \hat{\theta}_d)$ by:

$$(\hat{\theta}_f, \hat{\theta}_c) = \arg \min_{\theta_f, \theta_c} L_{sdet} + \alpha L_{DaC} - \lambda L_d^{sel} \quad (13)$$

$$\hat{\theta}_d = \arg \min_{\theta_d} \lambda (L_d^{sel} + L_d^{rej}) \quad (14)$$

where θ_f , θ_c , θ_d denote the parameters of the backbone network, category classifier, and the two domain discriminators respectively. α and λ are super parameters to balance those above losses.

5. Experiments

In this section, we will exhibit the results of our model and baselines with several datasets to demonstrate the effectiveness of our method. In experiments, VGG16 [31] is the backbone network of Faster R-CNN, and VGG16 is pre-trained in ImageNet.

5.1. Setting

Unless otherwise stated, all images of both train domain and target domain are resized as 600*800 pixels, if it less

then the size. For all experiments, we judge bounding box is correctly located if it's IOU higher than 0.5, and we show mean average precisions (mAP) of all models.

For our model, we set α and λ as 0.1 and 1.0 respectively. We optimize parameters of each model with stochastic gradient descent method (SGD), and the learning rate is set as 0.002 for the first 30k iterations and then reduces to 0.0002 for another 30k iterations, 0.00002 for last 30k iterations. The momentum scalar of equation 8 is 0.98 of all experiments. One batch is composed of a source domain image and a target domain image.

Baselines Our method is compared with following several baselines: (1) Faster R-CNN [28]. This model is trained only using source data, without any domain adaptive operation. (2) Adversarial discriminative domain adaptation (ADDA) [34]. ADDA is a classical adversarial domain adaptive model of classifier task, and just aligns image-level feature map under the setting of UDA object detection. (3) Domain adaptive Faster R-CNN [4]. DA Faster R-CNN aligns image-level features and instance-level features with adversarial domain adaptation. (4) Few-shot Adaptive Faster R-CNN (FSA Faster R-CNN) [36]. FSA Faster R-CNN randomly selects 9 bounding boxes to align the image-level feature map. (5) Strong-Weak Distribution Alignment (Strong-Weak DA) [29]. Strong-Weak DA proposed that strong alignment of high-level features can degrade model performance so that it does strong alignment of low-level features and weak alignment of high-level features.

5.2. Experiments

5.2.1 Transfer from Normal to Foggy Weather

Weather differences is a common domain shift between urban scene datasets. Therefore the first experiment focus on this scenario. We use *Cityscapes* [5] dataset as source domain and *Foggy Cityscapes* [30] as target domain.

Datasets The *Cityscapes* dataset is a popular benchmark in urban scene dataset. *Cityscapes* is photographed in nor-

mal weather, contains 8 category annotations: *bus*, *bicycle*, *car*, *mcycle*, *person*, *rider*, *train*, *truck*.

For the *Foggy Cityscapes* dataset, it is a synthetic dataset generated from *Cityscapes*, that simulate fog on real scenes. Foggy images are synthesized based on depth maps and real images, more details can be obtained in [30]. Following [4], we denote box envelope of instance mask in *Cityscapes* as bounding box annotations because *Cityscapes* is not made to detection.

Results Our model and other baselines results of the Normal-to-Foggy weather transfer experiment are shown in table 1. As summarized in table 1, ADDA aligns image-level feature map can improve Faster R-CNN which trains only with source domain by 5.0 mAP. Compare with ADDA, DA Faster R-CNN takes into account instance-level alignment and the relationship between image-level domain discriminator and instance-level discriminator, due to which DA Faster R-CNN further generates 2.7 mAP improvement. By solving the problem of different scale of objects, our model brings 3.1 mAP boost over DA Faster R-CNN. Further combining operation that rejects low transferability images, our model’s performance achieves 35.1 mAP. Strong-Weak DA [29] reduces negative transfer by reducing the weight of low transferability samples and get 34.3 mAP. And combining all components, our method enhances the detector by 17.1 mAP over Faster R-CNN or by 9.4 mAP over baseline DA Faster R-CNN model. To summarise, our model greatly outperforms than previous methods and get new state-of-art performance under the setting of unsupervised domain adaptive object detection.

5.2.2 Synthetic Data to Real

Manually annotate images with bounding boxes and categories is a time-consuming and expensive job. Fortunately, with the development of the computer graphics technique, labeled synthetic data is easily synthesized. So learning from synthetic data is meaningful and our second experiment is to compare our model and baselines in this scenario.

Datasets In this experiment, *SIM 10K* [18] is source domain dataset, and *Cityscapes* [5] is target domain dataset. *SIM 10K* dataset contains 10,000 synthetic images are rendered by computer game Grand Theft Auto V (GTA 5). For *SIM 10K* dataset, there are 58,701 bounding boxes of car in 10,000 images all of which are training sets. *Cityscapes* dataset contains around 5,000 accurately annotated training images and 500 validation set images which taken by a car-mounted camera. Although there are 8 categories are annotated in *Cityscapes* dataset, we only use one category (*car*) in this scenario, because only *car* class labeled in *SIM 10K* dataset.

	Mul	Rej	DaC	car AP
Faster R-CNN [28]				33.5
ADDA [34]				36.1
DA Faster R-CNN [4]				38.9
FS Faster R-CNN [36]				41.2
ours	✓			40.8
ours	✓	✓		43.2
ours	✓	✓	✓	43.4

Table 2. Quantitative results of our method and baselines on Synthetic-to-Real transfer scenario.

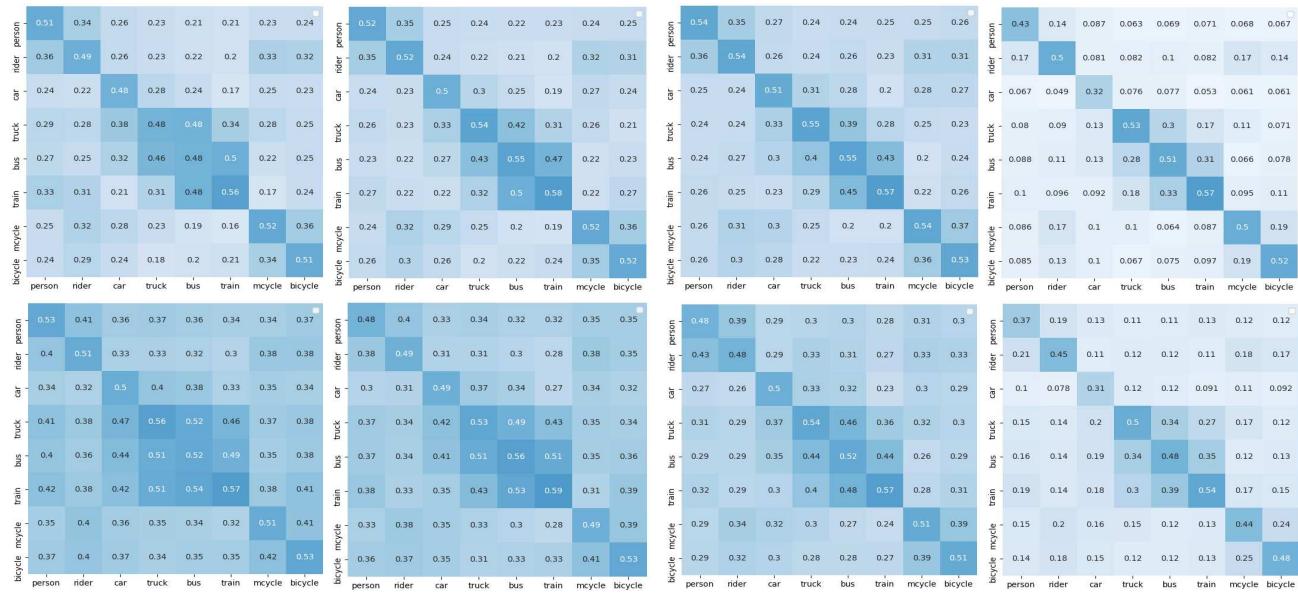
Results Table 2 shows the results of our model and baselines on the Synthetic-to-Real scenario. Similar to the Normal-to-Foggy weather transfer scenario, combining the Mul part and Rej part, our model outperforms than previous methods and achieves the best performance with all three constituents. Comparing 1.9 mAP improvement in the first scenario, it seems that the DaC constituent makes little difference to performance in this scenario. Perhaps it was because only one category is used in this experiment.

5.2.3 Visualization of DaC

To better explain what DaC does, we show the visualization of cosine similarity between instance-level feature vectors in each image and cosine similarity between every instance-level feature vector and all class central points (excluding background). Because models that did not combine DaC haven’t record central points after training. Before calculating cosine similarity, for all models, we estimate central points of each category with the momentum method as Equation 8. We set a momentum scale m as 0.98, and train with 3,000 steps, each step contains one source domain image and a target domain image, and every figure is calculated with 500 images.

We make this experiment in Normal-to-Foggy weather transfer scenario (*i.e.* *Cityscapes* \rightarrow *Foggy Cityscapes*). Faster R-CNN [28] and DA Faster R-CNN [4] are baseline models.

Results In figure 4, the first line are cosine similarity matrixes of the source domain and the second line are matrixes of the target domain. As we can see, for the Faster R-CNN model only trained with source domain, the cosine similarities between vectors and central of other classes are too high, some even higher than the similarity between vectors and the center of its class, which is bad for correctly classifying. What’s more, the difference between the source similarity matrix and target similarity matrix is large, which indicates to serious domain shift. For DA Faster R-CNN, due to adversarial domain adaptation, the difference between two domain similarity matrix is smaller. However, the simi-



Faster R-CNN

DA Faster R-CNN

ours w/o DaC

ours

Figure 4. Average cosine similarity between instances feature vectors and central point of 8 categories. Better viewed in color and zoom in for details.

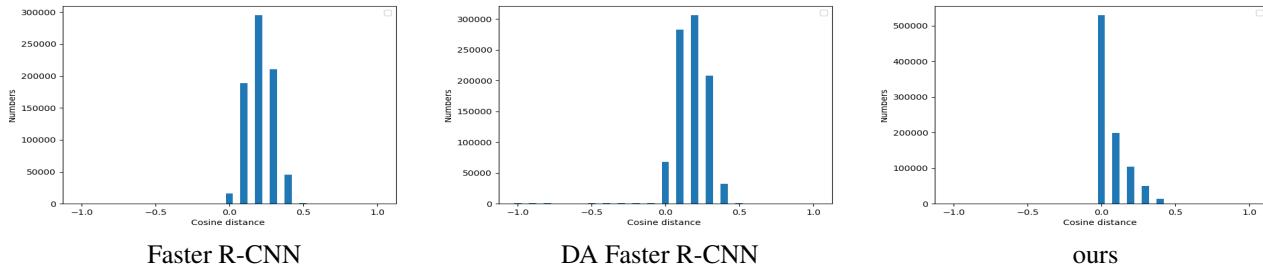


Figure 5. Average cosine similarity between instances feature vectors of each images of target domain.

larity score between feature vectors and the center of other classes are still too high. With multi-scale adaptation and sample selections, in our model (w/o DaC) the difference of two domains is smaller than DA Faster R-CNN. As for our model, similar to we noted above 4.4, the domain difference is even smaller, and feature vectors are far from other categories' center, which proves the cross-domain robustness of DaC.

As shown in figure 5, Faster R-CNN model's cosine distances between features are similar to DA Faster R-CNN, while our model's cosine distances remarkable higher. Just as stated above, DaC can diversify source domain and target domain distribution.

6. Conclusion

In this paper, we present a novel and effective model for cross-domain object detection. Our approach extends DA Faster R-CNN [28] by taking into account the multi-scale adaptation and reducing negative transfer. The proposed method solve these two problems by multi-channel

void convolutions and removing negative samples respectively. Moreover, diversifying-and-centering learning is derived to achieve better cross-domain robustness for cross-domain object detection. The proposed method is an end-to-end model and can robustly align source domain and target domain with unlabeled target images. Experiments on *Cityscapes*, *Foggy Cityscapes*, *SIM 10K* shown that our model outperforms the state-of-the-art for UDA object detection.

Acknowledgement

This work was supported partially by the Research Grant of Sun Yat-sen University and NSFC (No. 61906218).

References

- [1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *ICML*, pages 214–223, 2017.
- [2] A. Bendale and T. E. Boult. Towards open set deep networks. In *CVPR*, pages 1563–1572, 2016.

[3] Q. Cai, Y. Pan, C.-W. Ngo, X. Tian, L. Duan, and T. Yao. Exploring object relation in mean teacher for cross-domain detection. In *CVPR*, pages 11457–11466, 2019.

[4] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *CVPR*, June 2018.

[5] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, pages 3213–3223, 2016.

[6] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy. Optimal transport for domain adaptation. *PAMI*, 39(9):1853–1865, 2016.

[7] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010.

[8] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 17(1):2096–2030, 2016.

[9] S. Gidaris and N. Komodakis. Object detection via a multi-region and semantic segmentation-aware cnn model. In *ICCV*, pages 1134–1142, 2015.

[10] R. Girshick. Fast r-cnn. In *ICCV*, December 2015.

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, June 2014.

[12] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, pages 2066–2073, 2012.

[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014.

[14] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *ICCV*, pages 999–1006, 2011.

[15] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. In *NIPS*, pages 5767–5777, 2017.

[16] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask r-cnn. In *ICCV*, Oct 2017.

[17] L. P. Jain, W. J. Scheirer, and T. E. Boult. Multi-class open set recognition using probability of inclusion. In *ECCV*, pages 393–409. Springer, 2014.

[18] M. Johnson-Roberson, C. Barto, R. Mehta, S. N. Sridhar, K. Rosaen, and R. Vasudevan. Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? *arXiv preprint arXiv:1610.01983*, 2016.

[19] P. R. M. Júnior, R. M. de Souza, R. d. O. Werneck, B. V. Stein, D. V. Pazinato, W. R. de Almeida, O. A. Penatti, R. d. S. Torres, and A. Rocha. Nearest neighbors distance ratio open-set classifier. *ML*, 106(3):359–386, 2017.

[20] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017.

[21] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017.

[22] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014.

[23] M. Long, Z. Cao, J. Wang, and M. I. Jordan. Conditional adversarial domain adaptation. In *NIPS*, pages 1640–1650, 2018.

[24] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Unsupervised domain adaptation with residual transfer networks. In *NIPS*, pages 136–144, 2016.

[25] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Deep transfer learning with joint adaptation networks. In *ICML*, pages 2208–2217. JMLR. org, 2017.

[26] S. J. Pan and Q. Yang. A survey on transfer learning. *TKDE*, 22(10):1345–1359, 2009.

[27] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, June 2016.

[28] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015.

[29] K. Saito, Y. Ushiku, T. Harada, and K. Saenko. Strong-weak distribution alignment for adaptive object detection. In *CVPR*, pages 6956–6965, 2019.

[30] C. Sakaridis, D. Dai, and L. Van Gool. Semantic foggy scene understanding with synthetic data. *IJCV*, 126(9):973–992, 2018.

[31] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CS*, 2014.

[32] B. Sun, J. Feng, and K. Saenko. Return of frustratingly easy domain adaptation. In *AAAI*, 2016.

[33] A. Tarvainen and H. Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NIPS*, pages 1195–1204, 2017.

[34] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *CVPR*, pages 7167–7176, 2017.

[35] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

[36] T. Wang, X. Zhang, L. Yuan, and J. Feng. Few-shot adaptive faster r-cnn. In *CVPR*, pages 7173–7182, 2019.

[37] X. Wang, L. Li, W. Ye, M. Long, and J. Wang. Transferable attention for domain adaptation. In *AAAI*, 2019.

[38] Z. Wu, Y. Xiong, S. Yu, and D. Lin. Unsupervised feature learning via non-parametric instance-level discrimination. In *CVPR*, 2018.

[39] Z. Zhong, L. Zheng, Z. Luo, S. Li, and Y. Yang. Invariance matters: Exemplar memory for domain adaptive person re-identification. In *CVPR*, pages 598–607, 2019.