One-Shot Unsupervised Domain Adaptation for Object Detection

Zhiqiang Wan*, Lusi Li*, Hepeng Li*, Haibo He*, and Zhen Ni†
*Department of Electrical, Computer and Biomedical Engineering
University of Rhode Island, Kingston, USA
Email: {zwan, lli}@ele.uri.edu, {hepengli, haibohe}@uri.edu
†Department of Computer and Electrical Engineering and Computer Science
Florida Atlantic University, Boca Raton, USA
Email: zhenni@fau.edu

***Abstract*—The existing unsupervised domain adaptation (UDA) methods require not only labeled source samples but also a large number of unlabeled target samples for domain adaptation. Collecting these target samples is generally time-consuming, which hinders the rapid deployment of these UDA methods in new domains. Besides, most of these UDA methods are developed for image classification. In this paper, we address a new problem called one-shot unsupervised domain adaptation for object detection, where only one unlabeled target sample is available. To the best of our knowledge, this is the first time this problem is investigated. To solve this problem, a one-shot feature alignment (OSFA) algorithm is proposed to align the low-level features of the source domain and the target domain. Specifically, the domain shift is reduced by aligning the average activation of the feature maps in the lower layer of CNN. The proposed OSFA is evaluated under two scenarios: adapting from clear weather to foggy weather; adapting from synthetic images to real-world images. Experimental results show that the proposed OSFA can significantly improve the object detection performance in target domain compared to the baseline model without domain adaptation.**

***Index Terms*—Domain adaptation, object detection, deep learning**

I. Introduction

Deep neural network (DNN) based models have achieved great success in many applications [1]–[4]. These models are trained on a large amount of annotated data that are collected in a source domain. After the training process, these models are evaluated on a test set collected in a target domain. In general, it is assumed that the distributions of the source domain and the target domain are similar. However, in practice, this assumption is not always true. For example, the source domain data may be collected in the clear weather with good visibility while the target domain data may be collected in the foggy weather. The distribution discrepancy between the source domain and the target domain is called domain shift. Due to domain shift, the performance of these models may deteriorate sharply in the target domain.

To reduce domain shift and obtain good performance in the target domain, many unsupervised domain adaptation (UDA) algorithms have been proposed. These algorithms are unsupervised because they do not need any annotation for the target domain data. The existing UDA algorithms can be roughly divided into two categories: instance reweighting and feature matching. Instance reweighting algorithms [5], [6] aim to alleviate domain shift by reweighting the source domain instances according to their correlation with target domain instances. Feature matching methods [7]–[10] attempt to learn domain-invariant features to reduce the cross-domain distribution discrepancy.

Most of the existing UDA algorithms are developed for domain adaptive image classification. Less attention has been paid to object detection, which is more challenging than classification. In addition, these UDA algorithms need a large number of unlabeled target samples, which may not always be available. When a model is deployed to a new target domain, we may need to collect unlabeled target samples in this domain. This process can be time-consuming and labor-intensive, which hinders the rapid deployment of the models in real-world applications.

In this paper, we address a new problem called one-shot unsupervised domain adaptation for object detection. We only need one unlabeled target sample. This will significantly reduce the burden of collecting target samples and enhance the rapid deployment capability of the object detector in real-world applications. To the best of our knowledge, this is the first time this problem is investigated.

To solve this problem, we propose a one-shot feature alignment (OSFA) algorithm to align the low-level features of the source domain and the target domain. Specifically, the domain shift is reduced by aligning the average activation of the feature maps in the lower layer of convolutional neural network (CNN). Aligning low-level features can benefit domain adaptive object detection because these features are important for detecting the shape of the object. In addition, [11] have pointed out that the lower layer in a CNN can respond to low-level features, such as corners and edges. Source domain images and target domain images share common low-level features. However, due to domain shift, the illumination or visibility of the low-level features in the target images may be different from that in the source images. This difference makes it hard for CNN to recognize the low-level features in

This work was supported in part by the National Science Foundation under grant ECCS 1917275.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

the target domain. Therefore, it is essential to align low-level features to reduce domain shift.

The proposed OSFA algorithm is evaluated under two scenarios: adapting from clear weather to foggy weather; adapting from synthetic images to real-world images. In comparison with the baseline model without domain adaptation, the proposed algorithm significantly improves the object detection performance in the target domains.

The main contributions of this paper are as follows:

- We investigate a new problem called one-shot unsupervised domain adaptation for object detection. In this problem, only one unlabeled target sample is available for domain adaptation.
- We propose a novel algorithm to solve this problem by aligning the low-level features of the source domain and the target domain.
- We demonstrate the effectiveness of the proposed algorithm in numerous experiments under different domain adaptation scenarios.

II. Related Work

A. Object Detection

DNN can extract high-level from input data and has been widely used in many real-world applications [1], [3], [12]. With this learning capability, DNN has been widely used in object detection models [3], [13]–[16]. These object detection models can be roughly divided into two types: region proposal based model and regression/classification based model [17].

R-CNN [13] is the first model that brings deep CNN into object detection. Even though R-CNN outperforms the traditional object detection models with handcrafted image features, the training process of R-CNN is expensive in time and storage space because every region proposal should be processed by the whole CNN, and the extracted features are stored on the disk. To solve these problems, Fast R-CNN [14] is proposed to share the feature maps among region proposals. R-CNN and Fast R-CNN use an additional method, such as selective search, to produce the region proposal candidates. The region proposal generation is time-consuming for these two models. To further improve efficiency, Faster R-CNN [15] uses a Region Proposal Network (RPN) to produce high-quality region proposal candidates. RPN is nearly cost-free because it shares the feature maps with the detection network.

The above region proposal based object detection models contain several stages, including feature extraction, region proposal generation, classification, and bounding box regression. Unlike these models, regression/classification based models can directly map from an input image to the class probabilities and bounding box coordinates. Redmon *et al.* develop a regression/classification based framework called "You Only Look Once (YOLO)" [16]. The input image is divided into multiple grids, where each grid directly predicts several bounding boxes and their corresponding category probabilities. YOLO does not require region proposal generation and can perform object detection in real-time at 45 FPS. Since only the topmost feature layer is used in YOLO to generate bounding box predictions, it is hard for YOLO to detect small objects. In order to overcome this shortcoming, Liu *et al.* design an object detection model called "Single Shot MultiBox Detector (SSD)" [3]. SSD discretizes the output space of bounding boxes by utilizing several default anchor boxes with different aspect ratios and scales. SSD generates detection predictions from several feature maps with different resolutions such that it can detect objects with different sizes.

B. Unsupervised Domain Adaptation

A large number of existing UDA methods have been proposed for image classification and can be mainly grouped into two categories: instance reweighting and feature matching. Instance reweighting approaches attempt to identify the training source instances that are highly relevant to the target domain and exploit those reweighted instances to train a target model [5], [6]. Feature matching methods aim to align source and target domains by reducing the distribution gap of learned domain-invariant representations [18], [19]. Thereinto, transfer component analysis (TCA) [7] learns some shared transfer components in a Reproducing Kernel Hilbert Space (RKHS) to reduce the discrepancy across domains. While recent studies have shown that more transferable representative features can be learned by embedding UDA in the pipeline of deep representation learning [8], [9]. Currently, inspired by generative adversarial nets (GANs), adversarial UDA methods through an adversarial objective play the minimax game between the source and target domains to make them indistinguishable. For example, [10] present a unified framework, adversarial discriminative domain adaptation (ADDA), which learns two separate encoding networks that project source and target data into the same space by a domain-adversarial loss.

C. Domain Adaptation for Object Detection

Domain adaptive object detection is more challenging than domain adaptive classification because there may be multiple objects in the image, and we need to generate category and bounding box predictions for each object. Therefore, only few models [20]–[22] have been proposed for domain adaptive object detection. [20] introduce structural SVM to adapt the deformable part-based model (DPM) for pedestrian detection. This approach is semi-supervised because it needs some annotated target samples to adapt the structural model. [21] first train an object detection model on the source domain data. Then, this pre-trained model is fine-tuned on the artificially generated samples. This method is weakly-supervised because it requires image-level annotations of the target domain data to generate the artificial samples. [22] propose an unsupervised domain adaptation model for object detection by learning a domain-invariant RPN in the Faster R-CNN. Domain shift is reduced on both image level and instance level. The image-level and instance level domain adaptation are implemented by learning a domain classifier with adversarial training.

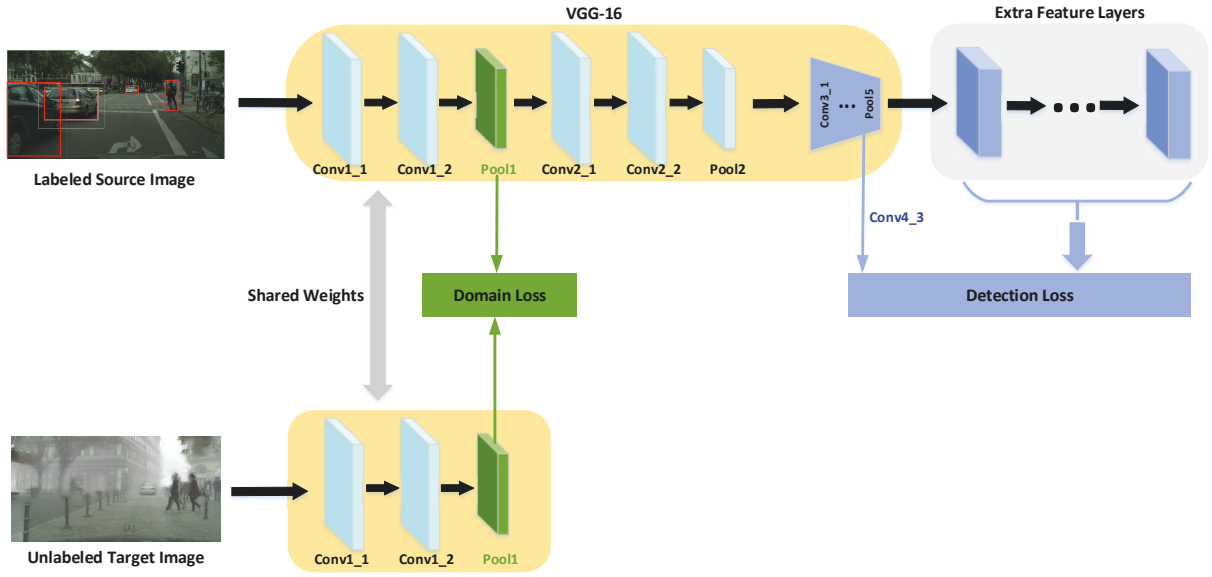

Fig. 1. The overall architecture of the proposed OSFA. Only one unlabeled target domain image is required for domain adaptation. A domain loss is calculated based on the output of the *Pool1* layer. During the training process, domain shift is reduced by minimizing this domain loss.

III. Preliminaries

In one-shot unsupervised domain adaptation for object detection, the images in the source domain are labeled with bounding boxes, and only one unlabeled image is available in the target domain. We use $\mathcal{D}_{\mathcal{S}} = \{(x_{S_1}, y_{S_1}), \cdots, (x_{S_n}, y_{S_n})\}$ to represent the samples in the source domain, where $x_{S_i} \in \mathcal{X}_{\mathcal{S}}$ denotes the input image, and $y_{S_i} \in \mathcal{Y}_{\mathcal{S}}$ denotes the bounding box annotation. We use $\mathcal{D}_{\mathcal{T}} = \{x_T\}$ to represent the sample in the target domain, where $x_T \in \mathcal{X}_{\mathcal{T}}$ denotes the input image.

The marginal distribution of source domain data X_S is denoted as $\mathcal{P}(X_S)$, and the marginal distribution of target domain data X_T is denoted as $\mathcal{Q}(X_T)$. Since there exists domain shift between the source domain and the target domain, these two distributions are not equal, i.e., $\mathcal{P}(X_S) \neq \mathcal{Q}(X_T)$. We want to determine a transformation function $f(\cdot)$ to map the source domain images and the target domain images into the same feature space such that the domain shift can be reduced. In this paper, a deep CNN is implemented as the transformation function. Its parameters are optimized by minimizing the detection loss and the discrepancy between $P(f(X_S))$ and $P(f(X_T))$. With this transformation function, the domain shift will be reduced, i.e., $P(f(X_S)) \approx P(f(X_T))$.

IV. One-Shot Unsupervised Domain Adaptation for Object Detection

In this section, we first introduce the motivations of low-level feature alignment and one-shot domain adaptation. Then, we provide the details about the proposed OSFA algorithm for one-shot unsupervised domain adaptive object detection.

A. Motivations

1) Low-Level Feature Alignment: Unlike the existing UDA algorithms that generally align high-level features of the CNN, we propose to reduce domain shift by aligning the low-level features. This idea is inspired by recent studies of visualizing and understanding the feature maps of CNN [11], [23]–[25] . These studies show that the upper layers of CNN respond to high-level features, such as the head of a dog, while the lower layers can be activated by low-level features, such as corners and edges. When a source domain image x_{S_i} and a target domain image x_T are inputted into a CNN, we can obtain the output of the upper layer as $f_H(x_{S_i})$ and $f_H(x_T)$. Since x_{S_i} and x_T may contain different objects, $f_H(x_{S_i})$ and $f_H(x_T)$ will be different. Thus, directly aligning the high-level features will not work. Nevertheless, x_{S_i} and x_T share common low-level features. Therefore, the output of the lower layer, $f_L(x_{S_i})$ and $f_L(x_T)$, will contain similar patterns. Thus, it is more reasonable to align the low-level features rather than the high-level features. In addition, aligning low-level features can benefit domain adaptive object detection because these features are critical for detecting the shape of the object.

2) One-Shot Domain Adaptation: In this paper, we propose to reduce domain shift by aligning the average activation of feature maps in the lower layer with only one target sample.

We use F to denote the activations of one feature map in the lower layer of a CNN. The activation of each unit in this feature map is denoted as $F_1, F_2, \cdots, F_K$. We assume $F_1, F_2, \cdots, F_K$ are independent and identically distributed random variables whose mean and variance are μ and σ^2, respectively. According to central limit theory, when K is large enough, the distribution of the average activation of this

feature map is close to the normal distribution with mean μ and variance $\frac{\sigma^2}{K}$ as Eq. (1).

$$\overline{F} = \frac{\sum_{k=1}^{k=K} F_k}{K} \rightarrow \mathcal{N}\left(\mu, \frac{\sigma^2}{K}\right). \quad (1)$$

When $K \rightarrow \infty$, the variance $\frac{\sigma^2}{K} \rightarrow 0$. In this case, this distribution will become $\delta\left(\overline{F} - \mu\right)$ where $\delta(.)$ denotes the Dirac delta function.

In the lower layer of a CNN, K is generally very large. For example, if the size of the input image is 512×512, and VGG16 is used to extract features, the value of K in *Pool1* layer is 65536. Therefore, the distribution in Eq. (1) will approach the Dirac delta function, and it is reasonable to use only one sample to estimate the mean value μ of this distribution.

B. Architecture of the Proposed Model

The overall architecture of the proposed OSFA is shown in Fig. 1. The input contains two images: the labeled source image and the unlabeled target image. The labeled source image is inputted into the network on the top to calculate the detection loss, and the unlabeled target image is fed into the network on the bottom. A domain loss is proposed to measure the domain shift between the source domain image and the target domain image. During the training process, the domain shift is reduced by minimizing this domain loss.

The network on the top is the well-known object detection model SSD. Since we align the low-level features of the source domain and the target domain, the proposed method can also be used in other object detection models. In SSD, the backbone network is a truncated VGG-16 (from *Conv1_1* to *Pool5*). On top of this backbone network, some convolutional layers are stacked to generate the detection prediction at multiple scales. Specifically, the detection predictions are generated from the following convolutional layers: *Conv4_3*, *Conv7*, *Conv8_2*, *Conv9_2*, *Conv10_2*, *Conv11_2*, and *Conv12_2*. The details of SSD can be found in [3]. The bottom network has three layers, i.e., *Conv1_1*, *Conv1_2*, and *Pool1*. Their parameters are shared with the truncated VGG-16 on the top.

By feeding the source image x_{S_i} into the top network, we can obtain the output of *Pool1* layer as $F^1_{S_i}$. Similarly, by inputting the target image x_T into the bottom network, we can obtain the output of *Pool1* layer as F^1_T. Then, we calculate the average activation of each feature map by applying global average pooling. Then, the domain loss L_{don} is calculated as Eq. (2).

$$L_{don} = \frac{1}{B}\sum_{i=1}^{B}\sum_{j=1}^{M_1}|G_j\left(F^1_{S_i}\right) - G_j\left(F^1_T\right)|, \quad (2)$$

where $G_j(\cdot)$ represents the global average pooling operation on the j_{th} feature map; $|.|$ calculates the element-wise absolute value; M_1 denotes the number of feature maps in *Pool1* layer; B represents the batch size.

Finally, we can calculate the total loss as Eq. (3).

$$L = L_{det} + \alpha L_{don}, \quad (3)$$

where L_{det} is the detection loss that measures the difference between the groundtruth and detection predictions; α is a coefficient. During the training process, the whole model in Fig. 1 is used. During the evaluation process, only the top network is used for object detection.

V. Experiments

In this section, we demonstrate the effectiveness of the proposed OSFA for one-shot domain adaptive object detection in numerous experiments under two scenarios. The first scenario is to adapt from clear weather to foggy weather. In this scenario, domain shift arises from different weather condition. The capability to adapt between different weather conditions is crucial for the real-world deployment of an object detector. The second scenario is to adapt from synthetic images to real-world images. In this scenario, the source domain images are generated by a simulation engine, and the target domain images are collected in real-world. The simulation engine can easily produce a large number of synthetic images with computer-generated bounding box annotation. The ability to adapt from synthetic images to real-world images can avoid the burden of manually collecting and annotating real-world images.

A. Experimental Setup

The training data consist of the annotated source domain images and one unlabeled target domain image. We use the original SSD model as a baseline for comparison. For this baseline model, we do not consider domain adaptation, and the model is trained with source domain data only. After the training process, the baseline model and the proposed OSFA are evaluated on the target domain data. In the following experiments, we use the mean average precision (mAP) as the evaluation metric, and the intersection over union (IOU) threshold is 0.5.

We choose the hyperparameters for SSD according to [3]. The value of the coefficient α is set as 0.001 for the following experiments. Stochastic gradient descent (SGD) with a momentum of 0.9 is used for the training. For the first 2k iterations, the learning rate is chosen as 5×10^{-4} to warm up the training. Then, the learning rate is set as 1×10^{-3} for the next 48k iterations. Then, for the next 10k iterations, the learning rate is reduced to 1×10^{-4}. For the final 10k iteration, the learning rate is further reduced to 1×10^{-5}.

B. Adapt from Clear Weather to Foggy Weather

In this section, we will demonstrate the effectiveness of the proposed OSFA under the first scenario, i.e., adapting from clear weather to foggy weather. The source domain images are from *Cityscapes* dataset [26], which is collected in clear weather condition. The target domain images are from *Foggy Cityscapes* dataset [27], which is obtained by simulating fog on the images in *Cityscapes*. Both *Cityscapes* and *Foggy Cityscapes* are collected for semantic segmentation

TABLE I
ADAPT FROM *Cityscapes* DATASET TO *Foggy Cityscapes* DATASET.

Methods	person	rider	car	truck	bus	train	motorcycle	bicycle	mAP
SSD [3]	22.0	27.6	37.0	16.7	28.2	9.9	19.8	30.1	23.9
img+ins align [22]	24.2	31.2	39.1	19.1	**36.2**	19.2	17.1	27.0	26.6
img+ins+cons [22]	**25.0**	31.0	40.5	22.1	35.3	**20.2**	20.0	27.1	27.6
OSFA (Pool2)	23.2	30.5	42.0	**23.8**	33.9	11.9	22.0	30.4	27.2
OSFA (Pool1)	23.6	**32.6**	**43.8**	22.9	35.4	14.7	**23.1**	**33.2**	**28.7**

and do not directly provide bounding box annotations. We follow the preprocessing step in [22] to get the bounding box annotation by taking the tightest rectangle of its instance. In *Cityscapes*, eight categories (person, rider, car, truck, bus, train, motorcycle, and bicycle) have instance labels. Thus, the experimental results are reported on these categories. During the training process, we use *Cityscapes*'s training set as source domain data and randomly select one unlabeled image from *Foggy Cityscapes*'s training set as the target domain data. After the training process, the baseline model and the proposed OSFA are evaluated on *Foggy Cityscapes*'s validation set.

The evaluation results of adapting from *Cityscapes* to *Foggy Cityscapes* are presented in Table I. Without domain adaptation, the mAP of the baseline SSD model is 23.9%. The average precisions (APs) of the eight categories are 22.0%, 27.6 %, 37.0%, 16.7%, 28.2%, 9.9%, 19.8%, and 30.1%, respectively. Compared to this baseline model, the proposed OSFA with *Pool2* layer for feature alignment can improve the mAP to 27.2%. With *Pool1* layer for feature alignment, the mAP can be further improved to 28.7%, and the APs of the eight categories are 23.6%, 32.6%, 43.8%, 22.9%, 35.4%, 14.7%, 23.1%, and 33.2%, respectively. We can see that the proposed OSFA outperforms the baseline model on mAP and across all the eight categories.

We also compare the proposed OSFA with two state-of-the-art domain adaptation models in [22]. It is worth noting that these two models require the whole training set of *Foggy Cityscapes* for domain adaptation while the proposed OSFA only needs one image for domain adaptation. We can see that the proposed OSFA with *Pool1* layer for feature alignment outperforms these two models. These results verify the effectiveness of the proposed OSFA in adapting from clear weather to foggy weather with only one unlabeled target image.

C. Adapt from Synthetic Images to Real-World Images

In this section, we will validate the effectiveness of the proposed OSFA under the second scenario, i.e., adapting from synthetic images to real-world images. The source domain images are from *SIM 10k* dataset [28]. The images and the corresponding bounding box annotations in this dataset are generated by Grand Theft Auto V (GTA V), which is a simulation engine. In *SIM 10k*, only *car* category is labeled. Thus, we report the evaluation results on this category. The target domain images are from three real-world datasets, i.e., *Cityscapes*, *PASCAL VOC2007*, and *PASCAL VOC2012*.

TABLE II
ADAPT FROM *SIM 10k* TO *Cityscapes*.

Methods	car AP
SSD [3]	35.1
img+ins align [22]	37.86
img+ins+cons [22]	38.97
OSFA (Pool2)	38.3
OSFA (Pool1)	38.6

TABLE III
ADAPT FROM *SIM 10k* TO *PASCAL VOC*.

Methods	VOC2007	VOC2012
SSD [3]	64.2	54.0
OSFA (Pool2)	69.1	57.5
OSFA (Pool1)	**71.2**	**58.0**

1) Domain Adaptation from SIM 10k to Cityscapes: During the training process, we use *SIM 10k* as the source domain data and randomly select one unlabeled image from *Cityscapes*'s training set as the target domain data. After the training process, the baseline and the proposed OSFA are evaluated on *Cityscapes*'s validation set.

The evaluation results of adapting from *SIM 10k* to *Cityscapes* are shown in Table II. Without domain adaptation, the AP of the baseline SSD model is 35.1%. Compared to this baseline, the proposed OSFA with *Pool2* layer for feature alignment can improve the AP to 38.3%. Furthermore, with *Pool1* layer for feature alignment, the AP is improved to 38.6%.

We also compare the proposed OSFA with two state-of-the-art domain adaptation models in [22]. These two models require the whole training set of *Cityscapes* for domain adaptation while the proposed OSFA only needs one image

TABLE IV
DOMAIN ADAPTATION BY ALIGNING FEATURES AT DIFFERENT LAYERS.

Methods	person	rider	car	truck	bus	train	motorcycle	bicycle	mAP
SSD [3]	22.0	27.6	37.0	16.7	28.2	9.9	19.8	30.1	23.9
Conv4_3	23.2	29.4	37.5	17.8	27.1	11.4	23.0	32.0	25.2
Conv7	23.6	31.6	38.5	15.0	28.2	9.1	18.2	32.2	24.6
Conv8_2	22.7	29.8	38.2	18.6	32.7	9.1	21.1	31.9	25.5
Conv9_2	22.7	30.7	38.3	18.7	29.7	9.7	19.6	30.8	25.0
Conv10_2	23.7	31.6	39.7	20.2	29.4	9.6	22.8	29.6	25.8
Conv11_2	24.2	31.9	38.5	19.0	29.6	9.3	20.3	31.9	25.6
Conv12_2	**24.6**	30.9	40.7	17.8	27.7	3.4	22.6	31.9	25.0
OSFA (Pool1)	23.6	**32.6**	**43.8**	**22.9**	**35.4**	**14.7**	**23.1**	**33.2**	**28.7**

for domain adaptation. We can observe that the performance of the proposed OSFA is close to the performance of these two models. These results demonstrate the effectiveness of the proposed OSFA in adapting from synthetic images to real-world images with only one unlabeled target image.

2) Domain Adaptation from SIM 10k to PASCAL VOC: In the following two experiments, we use *SIM 10k* as the source domain data and randomly select one unlabeled image from the training set of *VOC2007* or *VOC2012* as the target domain data. After the training process, the baseline model and the proposed OSFA are evaluated on *VOC2007*'s test set and *VOC2012*'s validation set.

The evaluation results of adapting from *SIM 10k* to *PASCAL VOC* are presented in Table III. Without domain adaptation, the APs of the baseline SSD model for these two experiments are 64.2% and 54.0%, respectively. In comparison with this baseline, the propose OSFA with *Pool2* layer for feature alignment can improve the APs to 69.1% and 57.5%, respectively. With *Pool1* layer for feature alignment, the APs for these two experiments can be further improved to 71.2% and 58.0%, respectively. These evaluation results demonstrate the proposed OSFA is effective in adapting to different target domains.

D. Ablation Studies

In this section, we first compare the performance of domain adaptation with different layers of CNN to show the advantage of the proposed low-level feature alignment. Then, we verify that the proposed OSFA can obtain good domain adaptation results regardless of the choice of the target sample. Finally, we investigate the robustness of the proposed OSFA to the coefficient α.

1) Feature Alignment with Different Layers: In the proposed OSFA, lower layer, *Pool1* layer, is used for feature alignment. In this section, we will compare the performance of low-level feature alignment with that of high-level feature alignment. We follow the experimental setup and the training process in Section "*Adapt from Clear Weather to Foggy Weather*" to train seven new models with upper layers (*Conv4_3*, *Conv7*, *Conv8_2*, *Conv9_2*, *Conv10_2*, *Conv11_2*, and *Conv12_2*) for feature alignment, respectively. In these experiments, we use *Cityscapes* as the source domain data and *Foggy Cityscapes* as the target domain data.

TABLE V
DOMAIN ADAPTATION WITH DIFFERENT TARGET SAMPLE.

Methods	$C \rightarrow F$	$S \rightarrow C$	$S \rightarrow 07$	$S \rightarrow 12$
SSD	23.9	35.1	64.2	54.0
OSFA (run1)	28.7	38.6	71.2	58.0
OSFA (run2)	29.9	38.5	71.0	57.5
OSFA (run3)	28.0	38.6	70.4	58.2
OSFA (run4)	28.5	37.7	71.5	57.2
OSFA (run5)	30.4	38.3	70.3	58.1
OSFA (average)	29.1	38.3	70.9	57.8

The results of these seven models are shown in Table IV and are compared with the baseline SSD model and the proposed OSFA with *Pool1* layer for feature alignment. The mAP of the baseline is 23.9%. The proposed OSFA with *Pool1* can improve the mAP to 28.7%. However, with the seven upper layers for feature alignment, the mAPs are 25.2%, 24.6%, 25.5%, 25.0%, 25.8%, 25.6%, and 25.0%, respectively. These results are only slightly better than the result of the baseline model and are worse than the result of the proposed OSFA. From these experiments, we can see that low-level feature alignment is better than high-level feature alignment.

2) Domain Adaptation with Different Target Sample: In the previous experiments, we randomly choose an unlabeled target image for feature alignment. In this section, we will analyze whether the choice of the target sample will affect the domain

TABLE VI
DOMAIN ADAPTATION WITH DIFFERENT COEFFICIENT α.

Methods	person	rider	car	truck	bus	train	motorcycle	bicycle	mAP
SSD [3]	22.0	27.6	37.0	16.7	28.2	9.9	19.8	30.1	23.9
OSFA ($\alpha = 0.0001$)	23.2	30.7	41.9	22.4	32.7	13.8	24.3	31.9	27.6
OSFA ($\alpha = 0.0005$)	24.2	32.4	43.6	22.7	35.8	16.2	24.2	31.3	28.8
OSFA ($\alpha = 0.001$)	23.6	32.6	43.8	22.9	35.4	14.7	23.1	33.2	28.7
OSFA ($\alpha = 0.005$)	22.5	31.8	45.0	19.7	36.6	20.1	25.7	33.7	29.4
OSFA ($\alpha = 0.01$)	22.2	32.6	43.2	24.9	32.7	19.6	23.0	31.4	28.7
OSFA ($\alpha = 0.05$)	22.5	32.3	44.5	18.7	35.3	23.0	21.7	30.4	28.5

adaptation performance. We conduct four domain adaptation experiments:

- Adapt from Cityscapes to Foggy Cityscapes ($C \rightarrow F$)
- Adapt from SIM 10k to Cityscapes ($S \rightarrow C$)
- Adapt from SIM 10k to PASCAL VOC2007 ($S \rightarrow 07$)
- Adapt from SIM 10k to PASCAL VOC2012 ($S \rightarrow 12$)

We run each experiment five times. In each run, we choose a different target sample. The results for these four experiments are shown in Table V. In $C \rightarrow F$, the mAP of the baseline is 23.9%. The mAPs of the five runs of the proposed OSFA are 28.7%, 29.9%, 28.0%, 28.5%, and 30.4%, respectively. The average value of these five runs is 29.1%. These results are all better than the result of the baseline. Similarly, in $S \rightarrow C$, the mAP of the baseline is only 35.1%. The mAPs of the five runs of the proposed OSFA are 38.6%, 38.5%, 38.6%, 37.7%, and 38.3%, respectively. The average value of these five runs is 38.3%. In $S \rightarrow 07$, the mAP of the baseline is 64.2%. The mAPs of the five runs of the proposed OSFA are 71.2%, 71.0%, 70.4%, 71.5%, and 70.3%, respectively. The average value of these five runs is 70.9%. In $S \rightarrow 12$, the mAP of the baseline is 54.0%. The mAPs of the five runs of the proposed OSFA are 58.0%, 57.5%, 58.2%, 57.2%, and 58.1%, respectively. The average value of these five runs is 57.8%. These results show that the proposed OSFA can obtain good domain adaptation results regardless of the choice of the target sample.

3) Robustness to Hyperparameter: In this section, we will investigate the robustness of the proposed OSFA to the coefficient α. We follow the experimental setup and the training process in Section "*Adapt from Clear Weather to Foggy Weather*" to train six OSFA models with different coefficient $\alpha = \{0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05\}$. The experimental results are shown in Table VI. With these different coefficients α, the corresponding mAPs are 27.6%, 28.8%,28.7 %, 29.4%, 28.7%, and 28.5%, respectively. These results are all better than the baseline model without domain adaptation. In addition, when the coefficient α varies from 0.0001 to 0.05, the mAPs are around 28.0%. These results show that the proposed OSFA is robust to the choice of coefficient α.

E. Insights and Discussions

1) Domain Shift Reduction: In this section, we will conduct experiments to show that the proposed OSFA succeeds in reducing domain shift. We follow the process in Section "*Adapt from Clear Weather to Foggy Weather*" to train a baseline SSD model without domain adaptation and the proposed OSFA with *Pool1* layer for feature alignment. Specifically, the baseline model is trained only with *Cityscapes* dataset. For the proposed OSFA, we use *Cityscapes* as the source domain data and randomly select one unlabeled image from *Foggy Cityscapes* as the target domain data. After the training process, we will calculate the domain shift for the baseline model and the proposed OSFA. The input for each model is a pair of images: x_{S_i} and x_{T_i}. x_{S_i} is an image from *Cityscapes* while x_{T_i} is from *Foggy Cityscapes* and is generated by simulating fog on x_{S_i}. Therefore, x_{S_i} and x_{T_i} will have the same objects. The only difference between them is that there is fog in x_{T_i}. This fog is the source of domain shift. We will validate whether the proposed OSFA can reduce this type of domain shift.

When x_{S_i} and x_{T_i} are inputted into the model, we can obtain the output of *Pool1* layer as $p^1_{S_i}$ and $p^1_{T_i}$. Then, we can calculate the domain shift in *Pool1* layer as Eq. (4).

$$D_{p^1} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} G_j\left(\left|\frac{p^1_{S_i} - p^1_{T_i}}{p^1_{S_i}}\right|\right), \tag{4}$$

where $|.|$ calculates the element-wise absolute value; $G_j(\cdot)$ represents the global average pooling operation over the j_{th} feature map; M is the number of feature maps in *Pool1* layer; N represents the number of samples in *Cityscapes*'s training set. Then, we calculate the domain shift reduction at *Pool1* layer as $\left(D^{base}_{p^1} - D^{OSFA}_{p^1}\right)/D^{base}_{p^1}$, where $D^{OSFA}_{p^1}$ represents the domain shift of the proposed OSFA; $D^{base}_{p^1}$ is the domain shift of the baseline model. Similarly, we can calculate the domain shift reduction for the other eight layers, including *Pool2*, *Conv4_3*, *Conv7*, *Conv8_2*, *Conv9_2*, *Conv10_2*, *Conv11_2*, *Conv12_2*.

The domain shift reduction for these nine layers are 67.39%, 18.39%, 14.85%, 26.77%, 17.61%, 2.90%, 0.81%, 14.58%,

and 17.03%, respectively. Since *Pool1* layer is used in the proposed OSFA for feature alignment, OSFA significantly reduces the domain shift at *Pool1* layer by 67.39%. For the remaining eight layers, even though they are not used for feature alignment, OSFA can still reduce the domain shift at these layers. These results demonstrate that the proposed OSFA is effective in reducing domain shift.

VI. Conclusion

In this paper, we investigated a new problem called one-shot unsupervised domain adaptation for object detection. Only one unlabeled target sample is available for domain adaptation. To solve this problem, we proposed an algorithm called one-shot feature alignment (OSFA) to align the low-level features of the source domain and the target domain. The proposed OSFA was evaluated under two domain adaptation scenarios. The experimental results showed that the proposed OSFA outperformed the baseline model without domain adaptation by a large margin. We further conducted ablation studies to compare the domain adaptation results when aligning different layers of CNN. The results verified the effectiveness of feature alignment with the lower layer of CNN. We also conducted experiments to show that the proposed OSFA can obtain good domain adaptation results regardless of the choice of the target sample. Finally, we performed experiments to demonstrate that the proposed OSFA succeeded in reducing domain shift.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778.

[2] Z. Wan and H. He, "Answernet: Learning to answer questions," *IEEE Transactions on Big Data*, pp. 1–1, 2018.

[3] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[4] Z. Wan, C. Jiang, M. Fahad, Z. Ni, Y. Guo, and H. He, "Robot-assisted pedestrian regulation based on deep reinforcement learning," *IEEE Transactions on Cybernetics*, pp. 1–14, 2018.

[5] J. Zhang, Z. Ding, W. Li, and P. Ogunbona, "Importance weighted adversarial nets for partial domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8156–8164.

[6] K. You, M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Universal domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2720–2729.

[7] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199–210, Feb 2011.

[8] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 136–144.

[9] P. O. Pinheiro, "Unsupervised domain adaptation with similarity learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8004–8013.

[10] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 2962–2971.

[11] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[12] Z. Wan, H. Li, H. He, and D. Prokhorov, "Model-free real-time ev charging scheduling based on deep reinforcement learning," *IEEE Transactions on Smart Grid*, vol. 10, no. 5, pp. 5246–5257, Sep. 2019.

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 580–587.

[14] R. Girshick, "Fast r-cnn," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 1440–1448.

[15] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 779–788.

[17] Z. Zhao, P. Zheng, S. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, 2019.

[18] J. Wen, N. Zheng, J. Yuan, Z. Gong, and C. Chen, "Bayesian uncertainty matching for unsupervised domain adaptation," *arXiv preprint arXiv:1906.09693*, 2019.

[19] Y. Shu, Z. Cao, M. Long, and J. Wang, "Transferable curriculum for weakly-supervised domain adaptation." AAAI, 2019.

[20] J. Xu, S. Ramos, D. Vzquez, and A. M. Lpez, "Domain adaptation of deformable part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 12, pp. 2367–2380, Dec 2014.

[21] N. Inoue, R. Furuta, T. Yamasaki, and K. Aizawa, "Cross-domain weakly-supervised object detection through progressive domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5001–5009.

[22] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. V. Gool, "Domain adaptive faster r-cnn for object detection in the wild," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 3339–3348.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.

[24] Z. Wan and H. He, "Weakly supervised object localization with deep convolutional neural network based on spatial pyramid saliency map," in *2017 IEEE International Conference on Image Processing (ICIP)*, Sep. 2017, pp. 4177–4181.

[25] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 2921–2929.

[26] 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 *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[27] C. Sakaridis, D. Dai, and L. Van Gool, "Semantic foggy scene understanding with synthetic data," *International Journal of Computer Vision*, vol. 126, no. 9, pp. 973–992, Sep 2018.

[28] 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?" in *IEEE International Conference on Robotics and Automation*, 2017, pp. 1–8.