Pattern Recognition 124 (2022) 108436

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Source data-free domain adaptation for a faster R-CNN

Lin Xiong[a], Mao Ye[a,*], Dan Zhang[a], Yan Gan[b], Yiguang Liu[c]

[a] School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, PR China
[b] College of Computer Science, Chongqing University, Chongqing 400044, PR China
[c] Vision and Image Processing Laboratory, School of Computer Science, Sichuan University, Chengdu 610065, PR China

ARTICLE INFO

Article history:
Received 29 August 2020
Revised 12 September 2021
Accepted 15 November 2021
Available online 19 November 2021

Keywords:
Source data-free
Object detection
Domain adaptation
Transfer learning

ABSTRACT

The existing domain adaptive object detection methods often need to carry a large number of source domain samples for domain adaptation, which is not realistic due to GPU limitations, privacy and physical memory in practical applications. To solve this problem, we propose a source data-free domain adaptive object detection method. Only unlabeled target domain data is used to optimize the source domain model so that it can work better in the target domain. Our method takes Faster R-CNN as baseline. Specifically, we first construct global class prototypes which will be updated in batch iteratively. Then based on the global class prototypes, more accurate pseudo-labels are generated for training the target model. In this way, the source and target domains are also implicitly aligned. Our contributions are 1) a prototype guided domain adaptation method which uses prototypes to mine the semantic category information without accessing the source dataset; 2) a scheme of iteratively updating global class prototype which can handle the class and sample imbalances in the training procedure and 3) a more accurate pseudo-label generation method combining semantic information and image information. On multiple public domain adaptive scenarios, our method achieves the state-of-the-art results in terms of accuracy compared with the Faster R-CNN model and some domain adaptive methods with source datasets.

1. Introduction

The problem of object detection is a very important problem in the field of computer vision. Given an image containing multiple objects, the task of the object detection problem is to locate the positions of these objects and classify each object. Many methods have emerged to solve the problem of object detection. These methods can be divided into two categories. The first category is called two-stage method such as R-CNN [1], Fast R-CNN [2], Faster R-CNN [3], R-FCN [4], Cascade R-CNN [5] and Mask R-CNN [6], etc. This type of method needs to first select the candidate region and then classify it. Another category is called one-stage method represented by YOLO [7]. In addition, there are other methods that also belong to the one-stage category, such as SSD [8], CornerNet [9], FCOS [10], CenterNet [11], RetinaNet [12] and memory based objector [13], etc. Compared with the two-stage method, the advantage of the one-stage method is that it can meet certain real-time requirements, but the disadvantage is that the accuracy is always reduced.

Although these object detection methods have already achieved good performance, they are all based on the assumption that both of the source and target domains obey the same distribution. However, the reality is that there always exists domain shift between the source and target domains. Figure 1 shows images in different domains. If we train an object detector in the source domain, and apply the obtained model to the target domain, the performance will drop sharply. Domain adaptation methods are applied for this situation. According to whether the data in target domain has a label, they can be divided into two categories, i.e., few-shot or unsupervised. Few-shot method assumes that some samples in the target domain have labels [14–17]; while for unsupervised case, there do not exist any labeled samples in the target domain.

For the problem of unsupervised domain adaptation of object classification, there are many works based on distribution matching, self-supervised learning or adversarial learning [18–20]. While for unsupervised domain adaptation of object detector, it was first raised by Domain Adaptive Faster R-CNN(DAF) [21]. Since then, many domain adaptive methods [22–24] have appeared one after another to solve the domain shift problem. This type of methods require both labeled source domain data and unlabeled target domain data to retrain object detector in the target domain. Keeping source domain data is not conducive to actual practical use because of privacy and computation resources.

* Corresponding author.
E-mail address: maoye@uestc.edu.cn (M. Ye).

https://doi.org/10.1016/j.patcog.2021.108436
0031-3203/

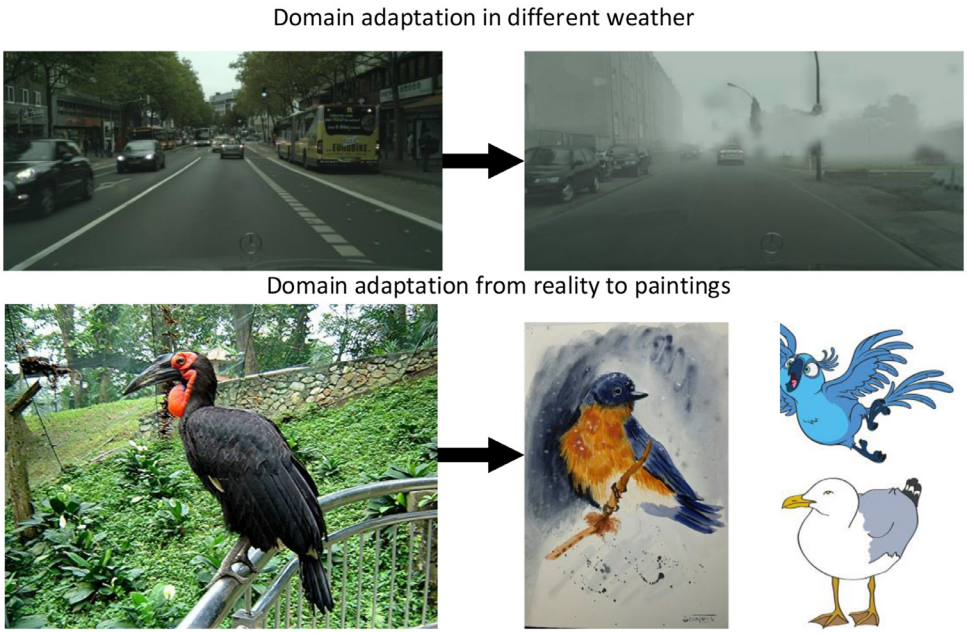


Fig. 1. Images of different styles. The images in the first row are from Cityscapes [25] and Foggy Cityscapes [26] respectively, which represent domain adaptation in different weather conditions. The left image in the second row is from Pascal VOC2007 [27], and three images on the right are from Clipart1k [28], Comic2k [28] and Watercolor2k [28] respectively. They represent the problem of domain adaptation from real images to multiple styles of painting.

The research of source data-free domain adaptation has first emerged for object classification problem in recent years [29–31]. These methods can be summed up with the following two points. The first point is using information maximization principle. If the distributions between two domains match, the prediction results in target domain by source classifier should not scatter, i.e. close to one-hot encodings. The second point is to generate pseudo-labels to target samples or target domain style images to retrain the source model so that it can work in the target domain. However, these methods cannot be applied to object detection problem directly. Source data-free domain adaptation for object detection is more challenging, because it needs to locate the object first and then classify it. For example, in the RPN network of Faster R-CNN, three hundred candidate regions in an image will be selected. These areas contain a lot of background and inaccurate object information. The fatal class and sample imbalance problem exists. Moreover, the context features are always used for better recognition results.

Under the assumption that the source datasets are not preserved and the tasks are the same for both of the target and source domains, we propose a prototype-based source data-free domain adaptation method based on the popular Faster R-CNN network. First, the global class prototypes are in batch iteratively constructed, which can better characterize the semantic information of each category in the target domain. Then these prototypes are combined with image features to generate more accurate pseudo-labels in target domain for self-supervised learning. And a divergence loss based on global class prototypes is defined to implicitly align the source and target domains. Furthermore, to guarantee the training stability, as the work in [30], we also require the model parameters do not change very much. Since we are not training a target model from scratch, only one epoch iteration is enough to obtain good performance in the target domain. In summary, our method has the following three advantages.

1) We proposed a novel prototype-based source data-free domain adaptation method without accessing the source datasets. The class prototype is somewhat similar to the cluster center and can describe the semantic category information to a certain extent. So it is reasonable to use prototype to mine some category information from the source model.

2) An iteratively updated scheme for global class prototype was proposed to save the category semantic information in the target domain. This semantic category information is not sensitive to the class imbalance problem for candidate regions and sample imbalance problem for each class.

3) Combining the semantic information of prototype and image features, a more accurate pseudo-labeling method was proposed. With these more accurate pseudo-labels, better global class prototypes will be updated and more discriminative features will be extracted.

Our method has achieved good results in multiple adaptation scenarios. Compared with Faster R-CNN, our method achieves a stable improvement of up to 7.2% in multiple senaries such as from SIM10k [32] to Cityscapes [25], from KITTI [33] to Cityscapes [25], etc. Our experimental results also show that our method can obtain competitive results compared with some domain adaptive object detection methods with source domain data.

2. Related works

2.1. Domain adaptation for object detection

The existing domain adaptive object detection methods with source datasets can be briefly divided into four types: 1) feature-level alignment; 2) data enhancement; 3) semi-supervised learning; and 4) robust learning.

Feature-level alignment is currently the mainstream method, which was first proposed by DAF [21]. This paper achieves the alignment between the source and target domains by minimizing the $\mathcal{H}$-divergence [34]. Adversarial learning makes the features of the source and target domains as close as possible [35]. The baseline is Faster R-CNN [3] and the alignments at the image level and instance level are performed respectively. After that, there emerged many feature-level-based works [23,24,36,37] to solve the problem of domain adaptive object detection. In summary, most of these methods use improved alignment or multi-layer alignment

techniques at the image or instance level. For example, Strong-Weak(SW) method [36] performs a weak-global alignment at the image level and a strong-local alignment in the middle of the feature extraction layer. For Multi-Aversarial Faster-RCNN(MAF) [37], the alignment is also done at the image level and instance level. The main difference is that there are multiple layers for alignment at the image level. The method in [23] combines attention map and MAF multi-layer alignments at image-level and at the instance level, where the prototypes of the source and target domains are used to achieve alignment.

The method based on data enhancement aims to use GANs or cycle-consistent Generative Adversarial Networks (CycleGAN) [38] to convert the source domain image into the target domain style for supervised training, which was first proposed in [28]. In addition to generating target domain styles from source domain data, [28] also generates pseudo-labels of the target domain and then fine-tunes the network. After that, different methods based on data enhancement such as [39–41] were proposed. For example, [39] not only uses CycleGAN to generate multiple intermediate domain images between the source and target domains, but also uses a multi-domain discriminator to learn invariant features between multiple domains to achieve feature-level alignment.

The semi-supervised learning category is to treat the domain adaptation problem as a semi-supervised problem. At present, we only find that [42] is based on the mean teacher in semi-supervised learning [42]. has a student model and a teacher model. The student model updates the parameters by minimizing cross-entropy loss and consistency loss. The parameters of the teacher model are the weighted exponential summation of the historical parameters of the student model, and finally a universal model applicable to the source domain and the target domain is learned.

The final category based on robust learning is proposed by Khodabandeh et al. [43]. This method learns a more accurate target domain pseudo-label, and then performs supervised training on the final network together with the labeled source domain data.

2.2. Source data-free domain adaptation

For source data-free domain adaptation problem, there do not exist many works which mainly focus on the task of object classification. Liang et al. [29] first proposes a method to solve this problem. Since the target domain has no labels, a clustering method is proposed to use pseudo-labeled target samples to retrain the source domain model. At the same time, if the features are aligned, then the correct classification probability on the target domain should be close to one-hot encodings. Thus information maximization principle is also used.

After that, the works [30,31] have emerged. Li et al. [30] generates images of each category in the target domain style, and then uses these images to retrain the source model. In addition, in order to prevent the model from deviating too much, the parameters of the model are also restricted, and the minimum entropy similar to Liang et al. [29] is used to make the classification probability is not scattered. The work done in [31] includes source data-free problem and universal categories problem. The proposed method first judges whether the category belongs to the known category in the source domain, and then uses the minimized entropy similar to Liang et al. [29] to make the classification probability close to one-hot encodings.

Compared with the source data-free adaptive object classification problem, the source data-free domain adaptation for object detection has the problems of class and data imbalances, which is more challenging. In order to solve these problems, our method proposes to use class prototypes, and also uses an iteratively updating scheme to solve class imbalance in a batch of training. Since we use class prototypes, the data imbalance impact for each class is also reduced.

3. Problem formulation

Suppose the labeled source domain data is $\mathcal{D}_S(X_S, Y_S)$ and unlabeled target domain data is $\mathcal{D}_T(X_T)$ respectively. X_S is the source domain image set and Y_S is the corresponding label set which have object locations and category information for each image. X_T is the target domain image set without labels. For the problem of source data-free domain adaptation of object detection, a mapping from source domain image to source domain label $f_S : X_S \rightarrow Y_S$ is given. The source and target data are in different distributions but the detection task is the same. The goal is to learn a mapping $f_T : X_T \rightarrow Y_T$ based on f_S and $\mathcal{D}_T(X_T)$ without accessing the source data set X_S.

For the mapping f_S, since most of domain adaptive methods are based on Faster R-CNN [3], for the convenience of comparison, we also use Faster R-CNN as the baseline model. Actually, the YOLO series can also be adapted after slight modification. Faster R-CNN includes three parts: feature extraction network, region proposal network (RPN) and classification network. For the Faster R-CNN network, as shown in the left part in Fig. 3, it is trained by minimizing $\mathcal{L}_{det}$ in Eq. (1),

$$\mathcal{L}_{det} = \mathcal{L}_{rpn\_cls} + \mathcal{L}_{rpn\_box} + \mathcal{L}_{roi\_cls} + \mathcal{L}_{roi\_box} \tag{1}$$

where $\mathcal{L}_{rpn\_cls}$ and $\mathcal{L}_{rpn\_box}$ are the foreground prediction loss and box regression loss from the RPN network, $\mathcal{L}_{roi\_cls}$ and $\mathcal{L}_{roi\_box}$ represent the category prediction loss and box precision regression loss obtained from the classification network.

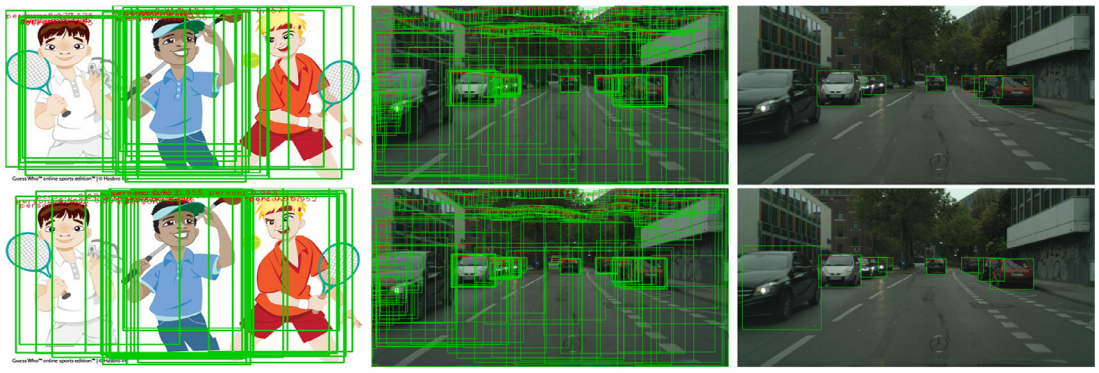
4. Problem analysis

4.1. Instance or image level alignment

In the previous approaches for domain adaptation of object detector, it is often necessary to align two domains at both of the image and instance levels, which is theoretically correct. However, the reality is that each image often contains multiple objects, and the position of each object is not fixed at all. So aligning two domains at the image level may not work as expected. In the source data-free domain adaptation case, where there does not exist the source domain data, transferring at the image level is more difficult.

Next we will show domain adaptation only at the instance level is also reasonable. Our method transfers Faster R-CNN with the froze RPN and box regression networks. We show the detection results obtained by the source model (see the first row of Fig. 2) and our adapted source model (see the second row of Fig. 2) on two transfer scenarios which are from Pascal VOC2007 to Multi-Paintings($P \rightarrow M$), and from SIM10k to Cityscapes($S \rightarrow C$) respectively. Figure 2a and b respectively show the detection results of the candidate regions in the $P \rightarrow M$ and $S \rightarrow C$ scenes. For the convenience of viewing, we keep only a small part of prediction boxes in the $P \rightarrow M$ scene. We can see that even if there are domain shifts between the source and target domains, all objects can be included in the candidate regions in different scenarios. The main reason that the source domain model does not work well in the target domain is due to the low classification scores of some object boxes, which will be eliminated by the NMS operation, as shown in the first row of Fig. 2c.

Therefore, in order to improve the performance of object detection in the target domain, we actually only need to improve the classification accuracy at instance level, so that the candidate regions can be screened to retain the correct object regions. In fact, for the domain adaptation problem of object detector with source



(a) $P \rightarrow M$ (Before NMS) (b) $S \rightarrow C$ (Before NMS) (c) $S \rightarrow C$ (After NMS)

Fig. 2. Visual comparison. The first and second rows show the detection results of the source model and our adapted model in the target domain respectively. $P \rightarrow M$ represents the transferring scenario from Pascal VOC2007 to Multi-Paintings, and $S \rightarrow C$ represents the transferring scenario from SIM10k to Cityscapes.

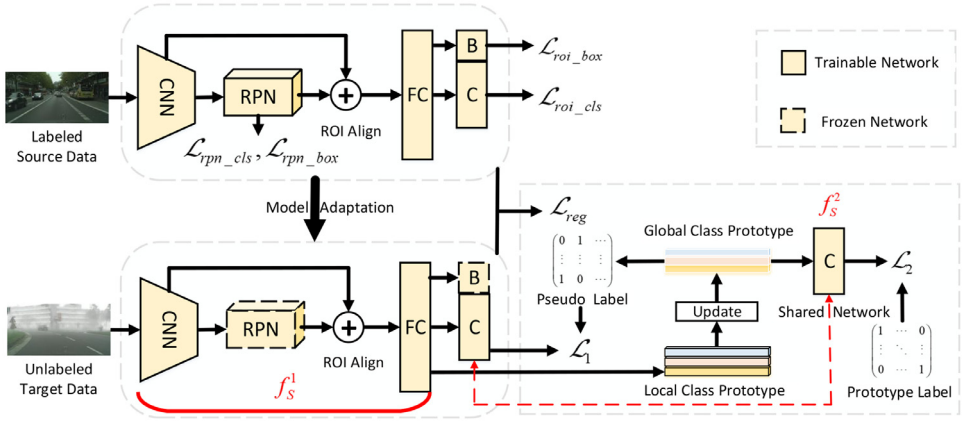


Fig. 3. An overview of the proposed method. In the left part, we first train the Faster R-CNN network with labeled source domain data. In the adaptation process, the trained source domain model is loaded for the target domain. Feature extraction network f_s^1 and classification layer f_s^2 will be adjusted by the unlabeled target domain data. The right part illustrates the proposed prototype-based method. The global class prototypes are iteratively updated by the local class prototypes which are produced in a batch. Then the global class prototypes are used to pseudo-label target samples to adapt the source domain model ($\mathcal{L}_1$) self-supervisely. Furthermore, we fed the prototype as feature to the classification network to align the source and target domains ($\mathcal{L}_2$). Finally, to guarantee smooth changes from the source domain model, a stability regularization term $\mathcal{L}_{reg}$ is added.

datasets, there are some approaches [22,42] which only operate domain adaptation at the instance level and also have achieved good results. Although we adjust the parameters of the feature extraction network to improve the classification accuracy, the candidate regions produced by the RPN network and the box regression network can still contain all our objects, as shown in the second row of Fig. 2a and b. And the score of correct object has been improved, so it can be retained after screening. The second row of Fig. 2c shows our final detection results. It can be seen that we have detected more objects than the baseline.

4.2. Prototype-based domain alignment

When the predecessors solved the domain adaptation problem, the general idea was to minimize the distribution divergence between the source domain $\mathcal{S}$ and target domain $\mathcal{T}$. Therefore, an $\mathcal{H}$-divergence [34] that measures distribution divergence between two domains needs to be defined first, which is shown in Eq. (2) [21].

$$d_{\mathcal{H}}(\mathcal{S},\mathcal{T}) = 2\left(1 - \min_{h\in\mathcal{H}} \left(err_{\mathcal{S}}(h(x)) + err_{\mathcal{T}}(h(x))\right)\right), \tag{2}$$

where x represents a sample from the source or target domain. $h : x \rightarrow \{0, 1\}$ is a domain classifier that predicts the source domain as 0 and the target domain as 1. $\mathcal{H}$ is assumed to be the set of possible domain classifiers. $err_{\mathcal{S}}$ and $err_{\mathcal{T}}$ are the prediction errors of h in the two domains. If the source domain data can be accessed, adversarial learning can be used to align the feature distributions between the source and target domains [35], as shown in Eq. (3) [21],

$$\min_{f} d_{\mathcal{H}}(\mathcal{S},\mathcal{T}) = \max_{f} \min_{h\in\mathcal{H}} \left(err_{\mathcal{S}}(h(x)) + err_{\mathcal{T}}(h(x))\right), \tag{3}$$

where f is the network that produces x.

With access to the source domain data, the feature distribution can be easily obtained from source domain data. But in the source data-free case, since source domain data is not accessible, Eq. (3) does not works. Inspired by the above $\mathcal{H}$-divergence, we redefine the distribution divergence between two domains. Denoting the target domain prototype of each category as $p_{\mathcal{T}}$, and the category classifier of source domain as $l_{\mathcal{S}} : p_{\mathcal{T}} \rightarrow \{0, 1, \ldots, \mathcal{C}\}$ respectively, we expect that the prediction of $p_{\mathcal{T}}$ by the classifier $l_{\mathcal{S}}$ is consistent with the corresponding category. Assuming that $\widetilde{\mathcal{H}}$ is the set of possible category classifiers, the divergence between two domains can be defined as follows,

$$d_{\widetilde{\mathcal{H}}}(\mathcal{S},\mathcal{T}) = err_{\mathcal{C}}(l_{\mathcal{S}}(p_{\mathcal{T}})), \tag{4}$$

where err_C is the prediction error of l_S. If the category for each prototype is the same as its predicted category with high-confidence, we consider that the alignment has been achieved implicitly.

Denoting the mapping that produces p_T as g, in order to align the source and target domains, we hope that p_T can minimize $d_{\widetilde{\mathcal{H}}}(\mathcal{S},\mathcal{T})$ as much as possible. At the same time, we also want to find an appropriate category classifier l_S to make $d_{\widetilde{\mathcal{H}}}(\mathcal{S},\mathcal{T})$ as small as possible. So the alignment problem of the two domains will be transformed into the following Eq. (5),

$$\min_g d_{\widetilde{\mathcal{H}}}(\mathcal{S},\mathcal{T}) = \min_g \min_{l_S \in \widetilde{\mathcal{H}}} err_C(l_S(p_T)). \tag{5}$$

5. The proposed method

5.1. Overview

Accroding to the analysis in Section 4, we propose a prototype-based adaptive method as shown in Fig. 3. First, a Faster R-CNN model is trained based on the labeled source domain data. The left part of Fig. 3 is a simplified diagram of Faster R-CNN structure. When the source model is adapting to the target domain, according to Section 4.1, we freeze the parameters of the RPN network and the box regression network, and only adjust the feature extraction network f_s^1 and the final classification layer f_s^2.

Our prototype-based method is shown in the right half of Fig. 3. For an unlabeled target domain image, the region features and their classification probabilities are obtained by the adapting source model. We first construct global class prototypes in batch incrementally that represent the semantic information of each category in the target domain (Section 5.2). Then, self-supervised learning technique is used, i.e., pseudo-labels for all regions in a batch of target samples are generated based on global prototypes and classification predictions. A cross-entropy loss $\mathcal{L}_1$ is defined to adapt the source model (Section 5.3). According to Section 4.2, the global class prototypes are used to implicitly align the source and target domains. A divergence loss $\mathcal{L}_2$ is defined to align the source and target domains (Section 5.4). Finally, a stability regularization term $\mathcal{L}_{reg}$ is added to guarantee that the adapted model is not too far from the source domain model (Section 5.5). In the end, the optimization process is summarized in Section 5.6.

Remark: By the analysis in Section 4.1, our method freezes RPN and box regression networks in the adaptation process. The fine-tuning of feature extraction network will affect the results of RPN, but this effect is very small, which are shown in Fig. 2. So our method is by sacrificing a little bit of position accuracy to a certain extent in exchange for the accuracy of the classification score, so as to improve the overall effect.

5.2. Prototype construction

The first step of our method is to construct global class prototypes. The global prototype can keep the semantic information of each category in target domain and can weaken the effect of the problem of class and data imbalances. Correspondingly, the local class prototype represents the semantic information of the category appeared in the current batch in the training procedure. Because a batch of images may not be able to include all categories, the local prototype may not be accurate. Therefore, we will use the knowledge of all images in the dataset to iteratively update the global prototype.

We first calculate the local class prototype $LP \in \mathbb{R}^{N_c \times N_d}$ in a batch by the following formula:

$$LP = W_{r2c} * F, \tag{6}$$

where $F \in \mathbb{R}^{N_r \times N_d}$ is the region features in a batch, $W_{r2c} \in \mathbb{R}^{N_c \times N_r}$ is the weight matrix mapping from the region to the category. N_c represents the number of categories, N_r represents the number of candidate regions in a batch, and N_d represents the dimension of the feature. The calculation formula for the value $W_{r2c}^{(i,j)}$ in the ith row and jth column is as follows

$$W_{r2c}^{(i,j)} = \frac{C(j,i)}{\sum_{j=1}^{N_c} C(j,i)}, \tag{7}$$

where $C \in \mathbb{R}^{N_r \times N_c}$ means the classification probability of each region, which can be obtained from the Faster R-CNN network.

The global class prototype $P \in \mathbb{R}^{N_c \times N_d}$ is iteratively updated. The current ((i)th batch) global prototype is obtained by the weighted summation of the current ((i)th batch) local prototype and the previous generation ($(i-1)$th batch) global prototype [23]. The update formula is as follows,

$$P^{(i)} = W_p^{(i)} \otimes LP^{(i)} + (1 - W_p^{(i)}) \otimes P^{(i-1)}, \tag{8}$$

in which $\otimes$ means channel-wise operation, and the initial global prototype is equal to the local prototype $P^{(0)} = LP^{(1)}$. $W_p^{(i)} \in \mathbb{R}^{N_c}$ represents the cosine similarity between the local prototype of the current generation ((i)th batch) and the global prototype of the previous generation ($(i-1)$th batch). The more similar the local prototype is, the greater the weight is. Conversely, the smaller the weight is when the current local prototype is not similar to the global prototype. We need to borrow more information of the global prototype of the previous generation. The similarity calculation formula is

$$W_p^{(i)}(k) = \left(\frac{LP^{(i)}(k)^T * P^{(i-1)}(k)}{\left\| LP^{(i)}(k) \right\| \left\| P^{(i-1)}(k) \right\|} + 1 \right) / 2, \tag{9}$$

for $1 \le k \le N_c$.

So far we have got the global class prototype. The following two modules will adapt the source model based on the constructed global prototypes.

5.3. Domain adaptation by pseudo-labeling

In this section, we will generate more accurate pseudo-labels for self-supervised training. We first calculate the cosine similarity $S \in \mathbb{R}^{N_r \times N_c}$ between the regional features and the global prototype of each category, which will predict which category each region belongs to from the perspective of semantic similarity. The calculation process is similar to the above cosine similarity,

$$S = \frac{F * P^T}{\|F\| * \|P\|}. \tag{10}$$

Although we can generate pseudo-labels for the target domain based on the prediction results of the above formula, such pseudo-labels are not comprehensive and accurate. Therefore, we need to combine the prediction probability of Faster R-CNN to calculate the pseudo-labels, so that the pseudo-labels will combine image information and semantic information at the same time. The semantic prediction result p_s and image prediction result p_c for the rth region is defined as follows,

$$p_s = S(r),\ p_c = C(r), \tag{11}$$

where C is the classification function of Faster RCNN in the previous section. The pseudo-label $\hat{y}$ for each candidate region is defined as follows:

$$\hat{y} = \arg\max_k \left(\beta * p_s(k) + p_c(k) \right) \tag{12}$$

where k is the specific category and β is used for balance so that the combined pseudo-label can make compromises between image information and semantic information. If the image prediction and semantic prediction results are the same, the result is naturally used as a pseudo-label. If there are differences, it is hoped

that the parameter β can enable the pseudo-label to obtain a value from the result of the higher score as much as possible.

After the pseudo-labels of the target domain are obtained, the standard cross-entropy loss can be used for training, as shown in the following formula:

$$\mathcal{L}_1 = \mathbb{E}_{(x_t,\hat{y}_t)\in(X_t,\hat{Y}_t)}[-\frac{1}{N_r}\sum_{i=1}^{N_r}\sum_{k=1}^{N_c}\mathbb{1}_{[k=\hat{y}_t^i]}\log(p(x_t^i))] \tag{13}$$

where x_t and $\hat{y}_t$ represent the images in a batch from target domain and the corresponding pseudo-labels respectively. x_t^i and $\hat{y}_t^i$ represent the ith candidate region and the corresponding pseudo-label. $\mathbb{1}_{[k=\hat{y}_t^i]}$ means to take 1 when $k=\hat{y}_t^i$, otherwise it takes 0.

Remark: Since only using high-confidence samples will cause source domain biased problem, our method uses all target samples in Eq. (13) which makes that the difficult samples can be concerned. Our idea is that as long as the overall quality of the pseudo-label is better than the original prediction result, the model will be optimized in a good direction.

5.4. Domain adaptation based on prototypes

Based on the divergence theory in Section 4.2, we will use the global class prototypes to align the source and target domains implicitly. If the source domain and target domain are aligned, the classification result of each class prototype should be consistent with its category.

Suppose the prediction results for all global class prototypes are $R \in \mathbb{R}^{N_c\times N_c}$. However, at the beginning of epoch, the prototypes of some categories are not completely formed. These prototypes have a negative impact on the domain alignment. So we use an attention matrix $A \in \mathbb{R}^{N_c\times N_c}$ in Eq. (14) to block the influence of the unsatisfied prototype on the network,

$$A = I(R > T), T = \frac{1}{N_c}, \tag{14}$$

where $I(\cdot)$ is the indication function. When the prediction result of the corresponding category is greater than the threshold, the standard backpropagation according to the loss of this category is performed; otherwise, the propagation is blocked. The threshold is determined by the number of categories in the dataset.

Since the prototype label of each category is itself, the prototype label $L \in \mathbb{R}^{N_c * N_c}$ of all categories will be an identity matrix. Therefore, our loss is defined as follows,

$$\mathcal{L}_2 = -\frac{1}{N_c}\sum_{k_1=1}^{N_c}\sum_{k_2=1}^{N_c}A(k_1,k_2)L(k_1,k_2)\log(R(k_1,k_2))), \tag{15}$$

where $R(k_1,k_2)$ represents the predicted probability that the prototype of the k_1th category belongs to the k_2th category. The function A decides which prototype can be backpropagated.

5.5. Stability regularization

Since there do not exist any labels in the training process, in order to prevent the model from deviating too far from the source domain model and losing its classification ability during the training process, inspired by Li et al. [30], we minimize parameter differences between the adjusted model and the source domain model, so that the model minimizes the previous losses with the smallest possible changes. This regularization term is

$$\mathcal{L}_{reg} = \|\theta - \theta_s\|, \tag{16}$$

where θ_s is the fixed parameters of source model, and θ is the parameters to be adjusted.

Algorithm 1 Pseudo code of prototype-based source data-free domain adaptation method.

Input: Pre-trained source model $f_S : X_S \to Y_S$, unlabeled target data $\mathcal{D}_T(X_T)$, batch size B, parameters $\alpha_1, \alpha_2, \lambda, \beta$
Output: An adapted model $f_T : X_T \to Y_T$

```
Load the Pre-trained source model
for each batch do
    X_T ← Sample(D_T, B)
    Build prototype
    Calculate local class prototype by Eq. (6)
    Update global class prototype by Eq. (8)
    More accurate pseudo-labels
    Calculate the similarity between the features of the candidate regions and global class prototype by Eq. (10)
    Combine semantic and image predictions to obtain pseudo-labels by Eq. (12)
    Calculate the cross-entropy loss by Eq. (13)
    Calculate attention by Eq. (14)
    Calculate the divergence loss based on prototypes by Eq. (15)
    Calculate the regularization term by Eq. (16)
    Optimize the model by minimizing Eq. (17)
end for
```

5.6. Optimization process

Our entire optimization process can be seen in Algorithm 1. Our total loss function is composed of three weighted losses as shown in the following formula,

$$\mathcal{L}_{total} = \alpha_1 * \mathcal{L}_1 + \alpha_2 * \mathcal{L}_2 + \lambda * \mathcal{L}_{reg}. \tag{17}$$

where α_1 and α_2 mainly depend on the initial accuracy of the source domain model in the target domain. The better the initial accuracy is, the more accurate the prototypes are. Thus the values of α_1 and α_2 should be larger. L_{reg} is used to replace weight decay, so the constraint coefficient λ for the model is always fixed for each experiment.

6. Experiments

We conduct source data-free domain adaptive object detection experiments based on the stable Faster R-CNN network with VGG16 backbone. The source domain data is not accessed, and only the source domain model is retained. In the adaptation process, we first load the source domain model, and then input the unlabeled target domain image to adapt it. Since there already exists a source model, it is enough to obtain good adaptation performance by traversing all the unlabeled target samples once.

6.1. Implementation details and datasets

6.1.1. Implementation details

Except for some of the parameters we mentioned, most of our parameter settings are based on the original Faster R-CNN. For all of experimental scenarios, all comparison methods are based on VGG16 backbone. The learning rate of all our experiments is $4e-6$, and we set weight decay to 0 because we already have a regularization term of L_{reg} to constrain the model parameter changes. In each experiment, the coefficient λ in front of the regularization term is set to 0.01, and the β used for balance is set to 10. As we mentioned, only one epoch is enough to complete the retraining. The specific number of iterations depends on the cardinality of target image set. Generally, only 1k to 2.5k iterations are needed when the batch size is 4.

Table 1
The composition of Multi-Paintings.

	Clipart1k	Comic2k	Watercolor2k	Total
Train	900	1800	1800	4500
Test	100	200	200	500

6.1.2. Datasets

In this section we will introduce all the datasets used in the experiments including Cityscapes [25], Foggy Cityscapes [26], KITTI [33], SIM10k [32], Pascal VOC2007 [27], Multi-Paintings [28].

Cityscapes contains 2975 training sets and 500 verification sets. The images in Cityscapes are all city street scenes under normal weather. There are eight categories of person, rider, car, truck, bus, train, motorcycle and bicycle.

Foggy Cityscapes contains images of cities in foggy weather synthesized from images in the Cityscapes dataset, which includes 2975 training sets and 500 verification sets. The categories of Foggy Cityscapes are exactly the same as that of Cityscapes.

KITTI contains 7481 labeled training set images, which are urban street scene images under a different camera setup. In order to transfer to the Cityscapes dataset, we only retain the car category in KITTI for experiments.

SIM10k contains 10,000 synthetic images obtained from the game GTAV. In order to transfer to the Cityscapes dataset, we only reserved the car category for experiments.

Pascal VOC2007 contains 20 categories of realistic images. We used its 5011 training set as the source domain, and retained 6 categories in order to transfer to Multi-Paintings.

Multi-Paintings is a painting dataset containing three domains. It is a combination of three datasets clipart1k [28], comic2k [28], and watercolor2k [28]. We used 6 public categories of the three datasets for experiments. Although the three datasets are all painting images, they come from different styles, namely clipart, comics, and watercolor. The 4500 training sets and 500 test sets of Multi-Paintings are obtained proportionally from these three datasets, as shown in Table 1.

6.2. Scenarios

In order to verify the effectiveness of our method, we conduct transferring experiments in 4 scenarios based on the above 6 datasets.

Adaptation in Adverse Weather (Cityscapes to Foggy Cityscapes): In order to verify the transferring effect of our method in different weather environments, we conduct a transferring experiment from Cityscapes to Foggy Cityscapes. The Cityscapes dataset containing normal weather city scenery is the source domain, and the Foggy Cityscapes dataset containing foggy city sceneries is the target domain. The parameters α_1 and α_2 are set to 0.1 and 0.05, respectively.

Learning from Synthetic Data (SIM10k to Cityscapes): We do a transferring experiment from SIM10k to Cityscapes to verify the transferring performance of our method from synthetic game images to real images. The source domain is SIM10k dataset, and the target domain is Cityscapes dataset. The AP of the public category car is the evaluation criterion. In this experiment, the parameters α_1 and α_2 are both set to 1.

Cross Camera Adaptation (KITTI to Cityscapes): In order to verify the transferring effect in different city street scenes, we conduct a transferring experiment from KITTI to Cityscapes. The KITTI dataset and Cityscapes dataset are used as the source and target domains respectively. The AP result of the public category car is used as the evaluation criterion. Since the number and categories of target domains in this experiment are consistent with the SIM10k to Cityscapes experiment, both α_1 and α_2 in this experiment are also set to 1.

Reality to Universal Paintings (Pascal VOC2007 to Multi-Paintings): In order to observe the transferring performance from real environment to multiple painting environments, we do an experiment from Pascal VOC2007 to Multi-Paintings. We select multiple datasets as the target domain to verify the stability of proposed method in universal environment. In this experiment, our source domain data is Pascal VOC2007. Because in actual transfer learning, the target domain itself may also have domain divergence. Therefore, we adapt the Multi-Paintings dataset containing multiple domains as the target domain. Pascal VOC2007 has 20 categories, and Multi-Paintings has 6 categories. We use their public 6 categories for transferring. The parameters α_1 and α_2 are set to 0.2 and 0.005 respectively.

6.3. Results

We report the performances of our complete method and two variants with only one module. For convenience, we use Ours(PLA) and Ours(DAP) to represent these two variants, i.e., with Pseudo-Labeling Adaptation module or Domain Adaptation with Prototypes module, which we proposed in Sections 5.3 and 5.4, respectively. As far as we known, no one has done work on source data-free domain adaptive object detection. So we will mainly compare our method with the baseline (Faster R-CNN without transferring). And some of state-of-the-art methods will also be compared that can access source domain data such as DAF [21], SW [36], SIR [44], FAFRCNN [45] and MAF [37].

6.3.1. Adaptation in adverse weather

In this experiment, in addition to comparing the baseline, we also compare our method with the first domain adaptive object detection method with source domain data DAF [21], as well as the method SW in [36]. We also do ablation experiments and report the results of each variant.

As can be seen from Table 2, our method has a 4.4% improvement compared to the baseline. The prediction results of two variants in each category are also better than the baseline (Source only). At the same time, our experimental results also exceed the two domain adaptive object detection methods with source domain data. So our method is effective for transferring in different weather.

6.3.2. Learning from synthetic data

In addition to comparing the baseline, we also compare a variety of methods with source domain data including DAF, MAF, FAFRCNN, SIR and SW in this experiment. We not only report the experimental results of our complete method but also the results of two variants.

As shown in Table 3 (S-C), our method far exceeds the baseline by up to 7.2%. And our method can surpass multiple methods with source domain data. Our two variant models have also shown good results, which can surpass most of methods.

6.3.3. Cross camera adaptation

In this experiment, the baseline, DAF and MAF are compared. We conduct ablation experiments and report the results of two variants.

As shown in Table 3 (K-C), our method improves the performance by 5% compared to the baseline, and also exceeds the results of DAF and MAF. The performances of two variants are also better than baseline and DAF.

Table 2
Cityscapes to Foggy Cityscapes. We report the mAP results for 8 categories and compare our method with the baseline, DAF and SW. The SW(L) in the table represents the local alignment in the SW paper. The best results are emphasized in bold.

Method	person	rider	car	truck	bus	train	mcycle	bicycle	mAP
Source only	25.83	33.30	35.16	12.98	26.38	9.11	18.98	32.31	24.26
DAF [21]	25.0	31.0	**40.5**	22.1	**35.3**	**20.2**	20.0	27.1	27.6
SW(L) [36]	27.5	**39.3**	36.2	**24.4**	34.3	5.4	**32.2**	23.7	27.9
Ours(PLA)	30.36	36.39	35.96	18.72	31.30	14.13	23.18	35.07	28.14
Ours(DAP)	30.08	35.47	35.60	15.67	31.64	9.49	22.90	35.07	26.99
Ours	**31.43**	38.58	38.98	17.01	31.29	13.24	23.50	**35.40**	**28.68**

Table 3
SIM10k to Cityscapes (S-C) and KITTI to Cityscapes (K-C). We report the AP of car in these two experiments. We compare our method with the baseline and multiple methods with source domain data including DAF, MAF, FAFRCNN, SIR and SW. The best results are emphasized in bold.

Method	S - C	K - C
Source only	34.96	36.69
DAF [21]	38.97	38.5
MAF [37]	41.1	41.0
FAFRCNN [45]	41.2	–
SW [36]	41.5	–
SIR [44]	40.3	–
Ours(PLA)	41.66	40.72
Ours(DAP)	41.12	39.21
Ours	**42.22**	**41.74**

6.3.4. Reality to universal paintings

Since the target domain dataset of this experiment contains three domains, no one has done this migration experiment, so we only compare our method with the baseline(Source only).

The performance comparisons of this experiment are shown in Table 4. Although we use a more challenging target domain dataset containing multiple subdomains, we can still see that our method has a mAP improvement of about 2.8% compared with baseline(Source only), and the prediction performance for almost every category is higher than the baseline.

6.4. Analysis

In the above experimental results, we have demonstrated the good results of our method and two variants. In this section, we will perform error analysis, parameter analysis, and visualization analysis to further verify the effectiveness of our method.

6.4.1. Error analysis

We take the SIM10k to Cityscapes experiment as an example to analyze the accuracy of the top ranked detections. The 3000 highest ranked prediction results are selected to analyze our method and the baseline (Source only). The detection results are classified into 3 categories according to the IOU size of the detection region and the ground-truth region, i.e., correct: $IOU > 0.5$; mislocalization: $0.1 < IOU < 0.5$ and background: $IOU < 0.1$ or others.

Our experimental results are shown in Fig. 4. Although the RPN network is frozen in the adaptation process, our method still reduces the number of false detections and backgrounds, and greatly increases the number of correct regions.

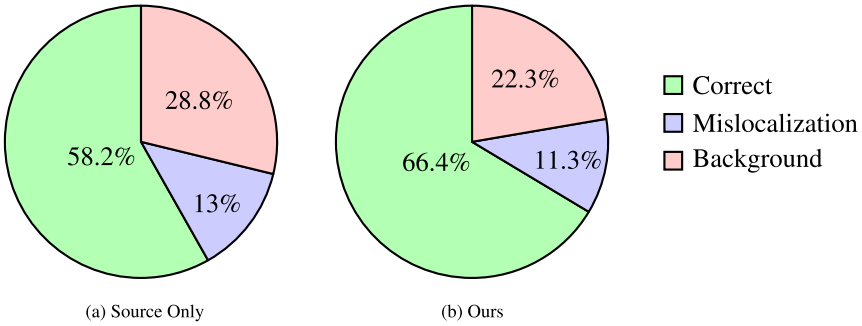
6.4.2. Parameters analysis

In this section, we analyze the parameters α_1, α_2 and β used in our method. In order to analyze parameter sensitivity, we take the SIM10k to Cityscapes experiment as an example to observe the performances by setting different parameter values in the experiment.

For the parameters α_1 and α_2, the strategy we adopt is to fix one parameter and constantly change the value of another parameter to observe the performance. Both parameters are set to the values 0.1, 0.2, 0.5, 1, 2, and 5. So we get a line chart as shown in Fig. 5. The blue line is the result of constant adjustment of the parameter α_2 when the parameter α_1 is fixed. The green line represents the result of fixing the parameter α_2 and changing the parameter α_1. The red line represents the result of the baseline (source only) and is used as a reference. It can be seen that our

Table 4
Pascal VOC2007 to Multi-Paintings. This is a challenging experiment in which there are multiple subdomains in a target domain. We only compare our method with the baseline. The best results are emphasized in bold.

Method	bicycle	bird	car	cat	dog	person	mAP
Source only	43.03	18.67	37.13	11.85	10.60	37.35	26.44
Ours(PLA)	42.32	**20.05**	39.06	11.38	14.36	**44.25**	28.57
Ours(DAP)	43.44	18.64	38.41	**11.93**	10.78	39.12	27.05
Ours	**43.59**	19.74	**42.61**	11.41	**14.37**	43.49	**29.20**



(a) Source Only (b) Ours

Fig. 4. Error analysis of the highest confident detections.

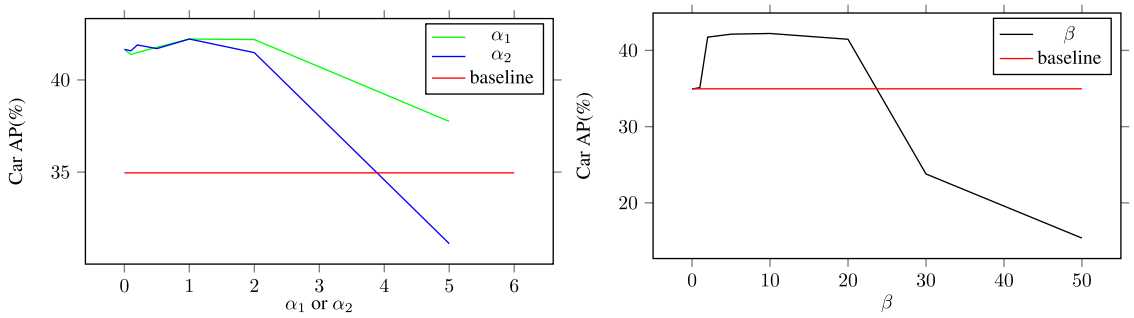
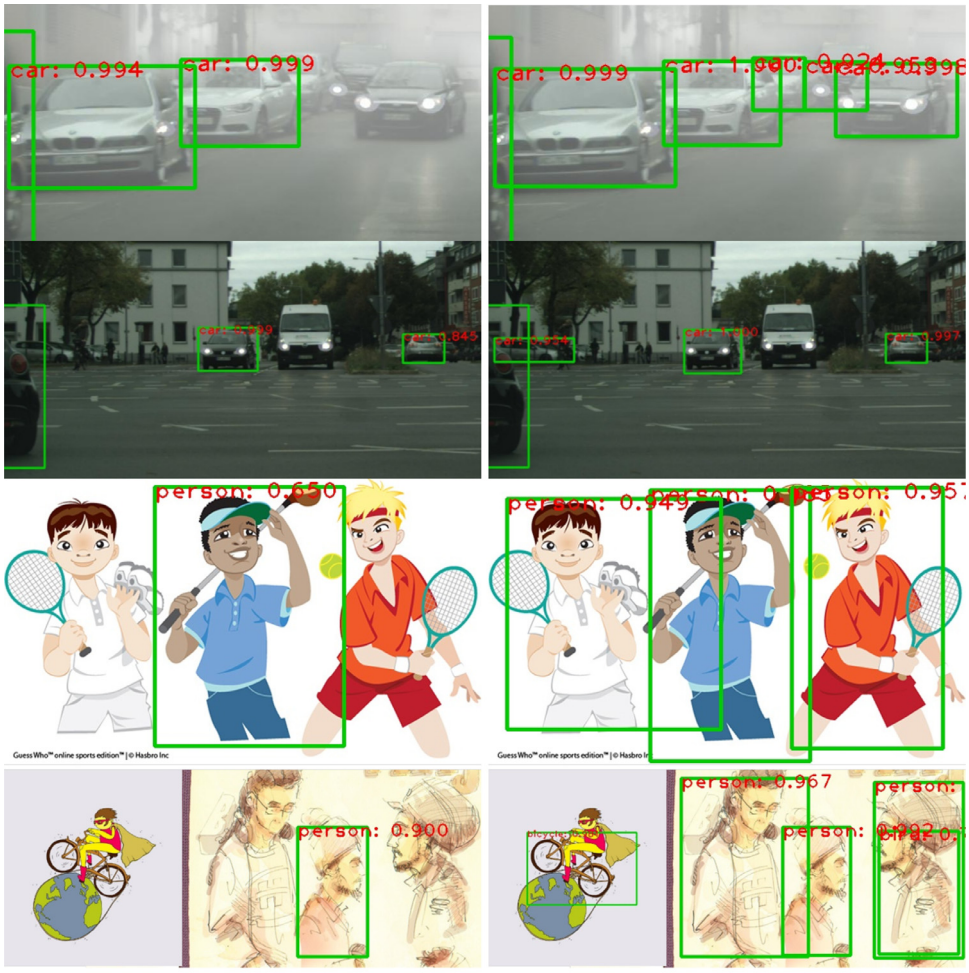


Fig. 5. Parameter Analysis. We take the SIM10k to Cityscapes experiment as an example. The figure on the left shows the performance of our complete method when only parameter α_1 or α_2 is changed. The figure on the right shows the result of Ours(PLA) variant when only parameter β is changed.



(a) Source Only (b) Ours

Fig. 6. Visual Analysis (Best viewed by zooming in). The results in the first row are from Cityscapes to Foggy Cityscapes experiment. The results in the second row are taken from KITTI to Cityscapes experiment. The results of the last two rows are the experimental results from Pascal VOC2007 to Multi-Paintings.

method is not sensitive to the changes of parameters, and it does not significantly decrease until the parameter reaches 5.

In order to better observe the effect of parameter β in Ours(PLA) variant, we only keep Ours(PLA) variant and constantly change the value of β to observe the performance. As shown in the line chart in Fig. 5, the value of β parameter in a large range can keep the result relatively stable, which is much higher than the baseline. When the β value approaches to 0, the pseudo-labels basically only depends on the original predictions, so the performance is similar to the baseline. When the value of β is large, the pseudo-labels are gradually dominated by semantic similarity. Since semantic similarity can only be used as an auxiliary, so the performance will be reduced. When β is set to 10, the semantic information and image information reach the optimal balance, and the final performance is the best.

6.4.3. Visualization analysis

In this section, we print the prediction box and score on the image. Since we have already shown the visualization effects of SIM10k to Cityscapes in Section 4.2, this section will show the performances of other three experiments.

Table 5
Failure cases compared with the latest methods that can access source data in the transferring scenarios from Cityscapes to Foggy Cityscapes (C-F), SIM10k to Cityscapes (S-C) and KITTI to Cityscapes (K-C). The best results are emphasized in bold.

Method	C - F	S - C	K - C
Source only	24.26	34.96	36.69
FAFRCNN [45]	31.3	41.2	–
SW [36]	34.3	41.5	–
SIR [44]	**40.2**	40.3	**46.5**
Ours	28.68	**42.22**	41.74

As shown in Fig. 6, the rendering of the baseline is on the left, and our rendering is on the right. The images in the first row show the experimental results from Cityscapes to Foggy Cityscapes. It can be seen that our method can frame more unclear cars. The images in the second row show the transferring experiment from KITTI to Cityscapes. It can be seen that we can still detect more hidden cars, and our scores are higher even if the objects are jointly detected. The last two lines show detection results from Pascal VOC2007 to Multi-Paintings. Our method can detect many objects while the baseline(Source only) cannot find them.

6.4.4. Failure cases

In order to further compare the latest works and analyze the performance of our method in different scenarios, we show the failure cases of our method when compared with the latest works SW [36], SIR [44], FAFRCNN [45] that can access source data. As shown in Table 5, we can see that the performance of our method is lower than the latest works in the transferring scenarios from Cityscapes to Foggy Cityscapes scene and the KITTI to Cityscapes scene, but the performance from the SIM to Cityscapes scene is still competitive. Because our method cannot access the source domain data, it is difficult to do feature alignment in such scenarios where the source domain and the target domain have a large gap.

7. Conclusions

In this paper, we proposed a prototype-based optimization method to solve the problem of source data-free domain adaptive object detection. Our method includes two modules: pseudo-labeling adaptation and domain adaptation with prototypes. We redefined the divergence between the source and target domains in a source data-free situation, and aligned the source and target domains based on the prototypes, and proposed a more accurate pseudo-label generation method. The global class prototypes are iteratively updated to save the semantic information of each category in the target domain, so as to reduce the effects of the imbalance problem of class and samples in object detection. Our scheme only needs to retrain the source domain model in one epoch, which greatly saves adaptation time. A lot of experiments are conducted in a variety of transferring scenarios and confirm the achieved improvements.

Compared with the existing methods that can access source domain data, our method does not need to access source domain data, which well protects the privacy of source domain data and reduces the use of computation resources. However, as most of self-supervised learning approaches, our method is also dependent on the initial accuracy of the source domain model applied to the target domain, which needs to be further improved in the future. In addition, it should be noted that our method does not completely exceed the existing methods that can access source domain data, so a new paradigm for source data-free domain adaptation is needed.

Declaration of Competing Interest

None.

Acknowledgment

This work was supported in part by the National Key R&D Program of China (2018YFE0203900), National Natural Science Foundation of China (61773093), Sichuan Science and Technology Program (2020YFG0476), and Important Science and Technology Innovation Projects in Chengdu (2018-YF08-00039-GX).

References

[1] R.B. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 580–587.
[2] R.B. Girshick, Fast R-CNN, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2015, pp. 1440–1448.
[3] S. Ren, K. He, R.B. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, IEEE Trans. Pattern Anal. Mach. Intell. 39 (2015) 1137–1149.
[4] J. Dai, Y. Li, K. He, J. Sun, R-FCN: object detection via region-based fully convolutional networks, in: Proceedings of the 30th International Conference on Neural Information Processing Systems, 2016.
[5] Z. Cai, N. Vasconcelos, Cascade R-CNN: delving into high quality object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 6154–6162.
[6] K. He, G. Gkioxari, P. Dollár, R.B. Girshick, Mask R-CNN, IEEE Trans. Pattern Anal. Mach. Intell. 42 (2020) 386–397.
[7] J. Redmon, S.K. Divvala, R.B. Girshick, A. Farhadi, You only look once: unified, real-time object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788.
[8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S.E. Reed, C.-Y. Fu, A.C. Berg, SSD: single shot multibox detector, in: Proceedings of the European Conference on Computer Vision (ECCV), 2016, pp. 21–37.
[9] H. Law, J. Deng, CornerNet: detecting objects as paired keypoints, Int. J. Comput. Vis. 128 (2020) 642–656.
[10] Z. Tian, C. Shen, H. Chen, T. He, FCOS: fully convolutional one-stage object detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 9626–9635.
[11] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, Q. Tian, CenterNet: keypoint triplets for object detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 6568–6577.
[12] T.-Y. Lin, P. Goyal, R.B. Girshick, K. He, P. Dollár, Focal loss for dense object detection, IEEE Trans. Pattern Anal. Mach. Intell. 42 (2020) 318–327.
[13] X. Li, M. Ye, Y. Liu, F. Zhang, D. Liu, S. Tang, Accurate object detection using memory-based models in surveillance scenes, Pattern Recognit. 67 (2017) 73–84.
[14] X. Li, M. Ye, Y. Liu, C. Zhu, Adaptive deep convolutional neural networks for scene-specific object detection, IEEE Trans. Circuits Syst. Video Technol. 29 (2019) 2538–2551.
[15] D. Li, J.-B. Huang, Y. Li, S. Wang, M.-H. Yang, Progressive representation adaptation for weakly supervised object localization, IEEE Trans. Pattern Anal. Mach. Intell. 42 (6) (2019) 1424–1438.
[16] L.A. Pereira, R. da Silva Torres, Semi-supervised transfer subspace for domain adaptation, Pattern Recognit. 75 (2018) 235–249.
[17] Y. Jia, J. Zhang, S. Shan, X. Chen, Unified unsupervised and semi-supervised domain adaptation network for cross-scenario face anti-spoofing, Pattern Recognit. 115 (2021) 107888.
[18] J. Liang, R. He, Z. Sun, T. Tan, Exploring uncertainty in pseudo-label guided unsupervised domain adaptation, Pattern Recognit. 96 (2019) 106996.
[19] Y. Chen, C. Yang, Y. Zhang, Y. Li, Deep conditional adaptation networks and label correlation transfer for unsupervised domain adaptation, Pattern Recognit. 98 (2020) 107072.
[20] W. Chen, H. Hu, Generative attention adversarial classification network for unsupervised domain adaptation, Pattern Recognit. 107 (2020) 107440.
[21] Y. Chen, W. Li, C. Sakaridis, D. Dai, L. Gool, Domain adaptive faster R-CNN for object detection in the wild, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 3339–3348.
[22] X. Zhu, J. Pang, C. Yang, J. Shi, D. Lin, Adapting object detectors via selective cross-domain alignment, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 687–696.
[23] Y. Zheng, D. Huang, S. Liu, Y. Wang, Cross-domain object detection through coarse-to-fine feature adaptation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 13763–13772.
[24] C. Xu, X. Zhao, X. Jin, X.-S. Wei, Exploring categorical regularization for domain adaptive object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 11721–11730.

[25] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, The cityscapes dataset for semantic urban scene understanding, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 3213–3223.
[26] C. Sakaridis, D. Dai, L.V. Gool, Semantic foggy scene understanding with synthetic data, Int. J. Comput. Vis. 126 (2018) 973–992.
[27] M. Everingham, L.V. Gool, C.K.I. Williams, J.M. Winn, A. Zisserman, The pascal visual object classes (VOC) challenge, Int. J. Comput. Vis. 88 (2009) 303–338.
[28] N. Inoue, R. Furuta, T. Yamasaki, K. Aizawa, Cross-domain weakly-supervised object detection through progressive domain adaptation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 5001–5009.
[29] J. Liang, D. Hu, J. Feng, Do we really need to access the source data? Source hypothesis transfer for unsupervised domain adaptation, in: Proceedings of the 37th International Conference on Machine Learning, 2020, pp. 6028–6039.
[30] R. Li, Q. Jiao, W. Cao, H.-S. Wong, S. Wu, Model adaptation: unsupervised domain adaptation without source data, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 9641–9650.
[31] J.N. Kundu, N. Venkat, R.V. Babu, et al., Universal source-free domain adaptation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 4544–4553.
[32] M. Johnson-Roberson, C. Barto, R.C.S. Mehta, S.N. Sridhar, K. Rosaen, R. Vasudevan, Driving in the matrix: can virtual worlds replace human-generated annotations for real world tasks? in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2017, pp. 746–753.
[33] A. Geiger, P. Lenz, C. Stiller, R. Urtasun, Vision meets robotics: the kitti dataset, Int. J. Rob. Res. 32 (2013) 1231–1237.
[34] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F.C. Pereira, J.W. Vaughan, A theory of learning from different domains, Mach. Learn. 79 (2009) 151–175.
[35] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A.C. Courville, Y. Bengio, Generative adversarial nets, in: Proceedings of the 27th International Conference on Neural Information Processing Systems, 2014.
[36] K. Saito, Y. Ushiku, T. Harada, K. Saenko, Strong-weak distribution alignment for adaptive object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 6949–6958.
[37] Z. He, L. Zhang, Multi-adversarial faster-RCNN for unrestricted object detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 6667–6676.
[38] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, T. Darrell, CyCADA: cycle-consistent adversarial domain adaptation, in: Proceedings of the International Conference on Machine Learning, 2018, pp. 1989–1998.
[39] T. Kim, M. Jeong, S. Kim, S. Choi, C. Kim, Diversify and match: a domain adaptive representation learning paradigm for object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 12448–12457.
[40] H.-K. Hsu, C.-H. Yao, Y.-H. Tsai, W.-C. Hung, H.-Y. Tseng, M. Singh, M.-H. Yang, Progressive domain adaptation for object detection, in: Proceedings of the IEEE Winter Conference on Applications of Computer Vision, 2020, pp. 749–757.
[41] A.L. Rodriguez, K. Mikolajczyk, Domain adaptation for object detection via style consistency, in: Proceedings of the British Machine Vision Conference (BMVC), 2019, pp. 204.1–204.14.
[42] Q. Cai, Y. Pan, C.-W. Ngo, X. Tian, L. yu Duan, T. Yao, Exploring object relation in mean teacher for cross-domain detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 11449–11458.
[43] M. Khodabandeh, A. Vahdat, M. Ranjbar, W.G. Macready, A robust learning approach to domain adaptive object detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 480–490.
[44] J. Chen, X. Wu, L. Duan, L. Chen, Sequential instance refinement for cross–domain object detection in images, IEEE Trans. Image Process. 30 (2021) 3970–3984.
[45] T. Wang, X. Zhang, L. Yuan, J. Feng, Few-shot adaptive faster R-CNN, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 7166–7175.

Lin Xiong received the B.E. degree from University of Electronic Science and Technology of China, Chengdu, China, in 2019. He is currently pursuing the M.S. degree at University of Electronic Science and Technology of China, Chengdu, China. His current research interests include machine learning, computer vision, and transfer learning.

Mao Ye received the B.S. degree from Sichuan Normal University, Chengdu, China, in 1995, and the M.S. degree from University of Electronic Science and Technology of China, Chengdu, China, in 1998 and Ph.D. degree from Chinese University of Hong Kong, China, in 2002, all in mathematics. He has been a short-time visiting scholar at University of Queensland, and University of Pennsylvania. He is currently a professor and director of CVLab with University of Electronic Science and Technology of China, Chengdu, China. His research interests include machine learning and computer vision. In these areas, he has published over 90 papers in leading international journals or conference proceedings. He has served on the editorial board of ENGINEERING APPLICATIONS OF ARTIFICIAL INTELLIGENCE. He was a co-recipient of the Best Student Paper Award at the IEEE ICME 2017.

Dan Zhang received the B.E. degree from Telecommunications Engineering from Southwest Minzu University, Chengdu, China, in 2017. She is currently pursuing the Ph.D. degree at University of Electronic Science and Technology of China, Chengdu, China. Her current research interests include machine learning, computer vision, and transfer learning.

Yan Gan received the B.S. degree and M.S. degree from Chongqing Normal University, Chongqing, China, in 2011 and 2015 respectively, and Ph.D. from University of Electronic Science and Technology of China, Chengdu, China, in 2020, all in computer science and technology. He is currently a postdoctoral researcher in college of computer science, Chongqing University, Chongqing, China. His current research interests include machine learning and computer vision. He is a reviewer for Engineering Applications of Artificial Intelligence and Neurocomputing.

Yiguang Liu received the M.S. degree from Peking University, Beijing, China, in 1998, and the Ph.D. degree from Sichuan University, Chengdu, China, in 2004. He was a Research Fellow, Visiting Professor, and Senior Research Scholar with the National University of Singapore, Singapore, Imperial College Lo ndon, London, U.K., and Michigan State University, East Lansing, MI, USA, respectively. He was chosen into the program for new century excellent talents of MOE in 2008, and chosen as a Scientifi c and Technical Leader in Sichuan Province in 2010. He is currently the Director of the Vision and Image Processing Laboratory and a Professor with the School of Computer Science, Sichuan Unive rsity. He has co-authored over 100 international journal and conference papers, and a chapter of the book entitled Computational Intelligence and Its Applications (H.K.Lam). His current research interests include computer vision and image processing, pattern recognition, and computational intelligence. Dr. Liu is a Reviewer of Mathematical Reviews of the American Mathematical Society.