Progressive Domain Adaptation for Object Detection

Han-Kai Hsu[1] Wei-Chih Hung[1] Hung-Yu Tseng[1] Chun-Han Yao[2]
Yi-Hsuan Tsai[3] Maneesh Singh[4] Ming-Hsuan Yang[1,5]

[1]University of California, Merced [2]University of California, San Diego [3]NEC Laboratories America
[4]Verisk Analytics [5]Google

Abstract

Recent deep learning methods for object detection rely on a large amount of bounding box annotations. Collecting these annotations is laborious and costly, yet supervised models do not generalize well when testing on images from a different distribution. Domain adaptation provides a solution by adapting existing labels to the target testing data. However, a large gap between domains could make adaptation a challenging task, which leads to unstable training processes and sub-optimal solutions. In this paper, we propose to bridge the domain gap with an intermediate domain and then progressively solve easier adaptation subtasks. Experimental results show that our method performs favorably against the state-of-the-art method in terms of the model test performance on the target domain.

1. Introduction

Object detection is an important computer vision task that aims to localize and classify objects in the images. Recent advancement in deep neural networks has brought significant improvement to the performance of object detection [7, 18, 15, 16, 17, 12]. However, such deep models usually require a large-scale annotated dataset for supervised learning and do not generalize well when the training and testing domains are different. For instance, the domains can differ in sceneries, weather, lighting conditions and the image appearance with respect to the camera being used. Such domain discrepancy or domain-shift causes unfavorable model generalization issues. Although adding additional training data from the target domain can improve the performance, collecting annotations is usually time-consuming and labor-intensive.

Unsupervised domain adaptation methods aim to solve the domain-shift problem without using ground truth labels in the target domain. Given the source domain annotations, the objective is to align the source and target feature distributions in an unsupervised manner, so that the model can generalize to the target data. Numerous methods are developed in the context of image classification [23, 13, 14, 20, 8, 22, 5, 1], while fewer efforts have been made on more complicated tasks such as semantic segmentation [10, 21] and object detection [9, 2, 11]. In fact, such domain adaptation tasks are quite challenging as there usually exists a significant gap between source and target domains.

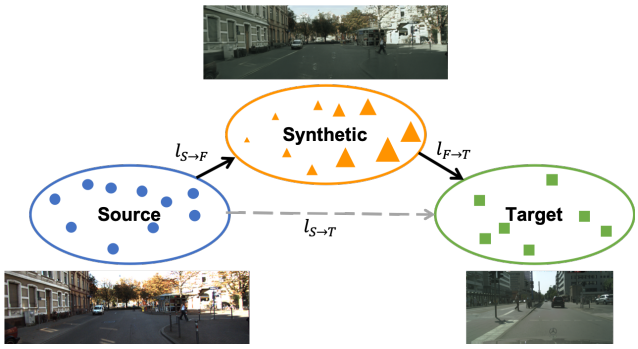

Figure 1. An illustration of our progressive adaptation method. Conventional domain adaptation aims to solve domain-shift problem from source to target domain, which is denoted as $l_{\mathbb{S}\to\mathbb{T}}$. We propose to bridge this gap with an intermediate synthetic domain that allows us to gradually solve separate subtasks with smaller gaps (shown as $l_{\mathbb{S}\to\mathbb{F}}$ and $l_{\mathbb{F}\to\mathbb{T}}$). In addition, we treat each image in the synthetic domain unequally based on its quality with respect to the target domain, where the larger size in yellow triangle stands for larger weights (i.e., the closer to the target, the higher of the weight).

In this paper, we aim to ease the efforts in aligning the different domains. Inspired by [8] that resolves the domain-shift problem via aligning intermediate feature representations, we utilize an intermediate domain that lies between the source and target domain, and hence avoid direct mapping across the two distributions with a significant gap (as

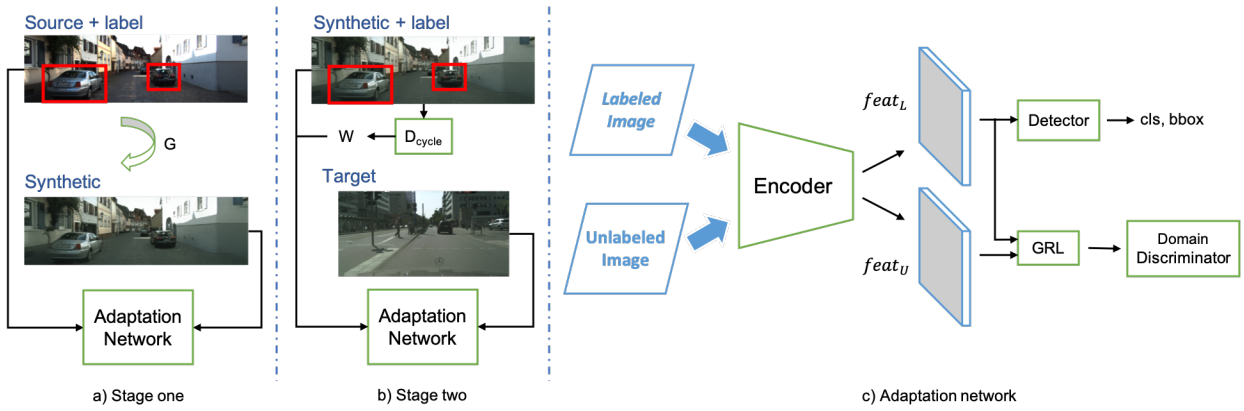

Figure 2. The proposed progressive adaptation framework. The algorithm includes two stages of adaptation as shown in a) and b). In a), we first transform source images to generate synthetic ones by using the generator G learned via CycleGAN [24]. Afterward, we use the labeled source domain and perform first stage adaptation to the synthetic domain. Then in b), our model applies a second stage adaptation which takes the synthetic domain with labels inherited from the source and aligns the synthetic domain features with the target distribution. In addition, a weight w is obtained from the discriminator D_{cycle} in CycleGAN to balance the synthetic image qualities in the detection loss. The overall structure of our adaptation network is shown in c). Labeled and unlabeled images are both passed through the encoder network E to extract CNN features $feat_L$ and $feat_U$. We then use them to: 1) learn supervised object detection with the detector network from $feat_L$, and 2) forward both features to GRL and a domain discriminator, learning domain-invariant features in an adversarial manner.

illustrated in Figure 1).

We conduct experiments on multiple domain discrepancy issues such as weather changes and camera differences. With the proposed progressive adaptation , we show that our method performs favorably against the state-of-the-art algorithm in the target domains. The main contributions of the work are summarized as follows: 1) we introduce an intermediate domain in the proposed domain adaptation framework to achieve progressive feature distribution alignment for object detection, 2) we develop a weighted task loss during domain alignment based on the importance of the samples in the intermediate domain, and 3) we conduct extensive adaptation experiments under various object detection scenarios and achieve state-of-the-art performance.

2. Progressive Domain Adaptation

We propose to decompose the domain adaptation problem into two smaller subtasks, bridged by a synthetic domain sitting in between the source and target distribution. We denote the source, synthetic, and target domains as $\mathbb{S}$, $\mathbb{F}$ and $\mathbb{T}$, respectively. The conventional adaptation from a labeled domain $\mathbb{S}$ to the unlabeled domain $\mathbb{T}$ is denoted as $\mathbb{S} \rightarrow \mathbb{T}$, while the proposed adaptation subtasks are expressed as $\mathbb{S} \rightarrow \mathbb{F}$ and $\mathbb{F} \rightarrow \mathbb{T}$. An overview of the proposed progressive adaptation framework is shown in Figure 2. We discuss the details of the proposed adaptation network and progressive learning in the following sections.

2.1. Adaptation in the Feature Space

In order to align distributions in the feature space, we propose a deep model which consists of two components: a detection network and a discriminator network for feature alignment via adversarial learning.

Detection Network. We adopt the Faster R-CNN [18] framework for the object detection task, where the detector has a base encoder network E to extract image features. Given an image $\mathbf{I}$, the feature map $E(\mathbf{I})$ is extracted and then fed into two branches: Region Proposal Network (RPN) and a Region of Interest (ROI) classifier. We refer to these branches as the detector, which is shown in Figure 2. To train the detection network, the loss function $\mathcal{L}_{det}$ is defined as:

$$\mathcal{L}_{det}(\mathbf{I}) = \mathcal{L}_{rpn} + \mathcal{L}_{cls} + \mathcal{L}_{reg}, \tag{1}$$

where $\mathcal{L}_{rpn}$, $\mathcal{L}_{cls}$, and $\mathcal{L}_{reg}$ are the loss functions for the RPN, classifier and bounding box regression, respectively.

Domain Discriminator. To align the distributions across two domains, we append a domain discriminator D after the encoder E. The main objective of this branch is to classify whether the feature $E(\mathbf{I})$ comes from the source or the target domain. Through this discriminator, the probability of each pixel belonging to the target domain is obtained as $\mathbf{P} = D(E(\mathbf{I})) \in \mathbb{R}^{H \times W}$. We then apply a binary cross-entropy loss to $\mathbf{P}$ based on the domain label d of the input image, where images from the source distributions are given the label $d = 0$ and the target images receive label $d = 1$. The discriminator loss function $\mathcal{L}_{disc}$ is formulated as:

$$\mathcal{L}_{disc}(E(\mathbf{I})) = -\sum_{h,w} d \log \mathbf{P}^{(h,w)} + (1-d) \log(1 - \mathbf{P}^{(h,w)}). \tag{2}$$

Adversarial Learning. Adversarial learning is achieved using the Gradient Reverse Layer (GRL) proposed in [4] to learn the domain-invariant feature $E(\mathbf{I})$. GRL is placed in between the discriminator and the detection network, only affecting the gradient computation in the backward pass. During back-propagation, GRL negates the gradients that flow through. As a result, the encoder E will receive gradients that force it to update in an opposite direction which maximizes the discriminator loss. This allows E to produce features that fools the discriminator while D tries to distinguish the domain. For the adaptation task $\mathbb{S} \rightarrow \mathbb{T}$, given source images $\mathbf{I}_{\mathbb{S}}$ and target images $\mathbf{I}_{\mathbb{T}}$, the overall min-max loss function of the adaptive detection model is defined as the following:

$$\min_E \max_D \mathcal{L}(\mathbf{I}_{\mathbb{S}}, \mathbf{I}_{\mathbb{T}}) = \mathcal{L}_{det}(\mathbf{I}_{\mathbb{S}}) + \lambda_{disc}\big[\mathcal{L}_{disc}(E(\mathbf{I}_{\mathbb{S}})) + \mathcal{L}_{disc}(E(\mathbf{I}_{\mathbb{T}}))\big], \quad (3)$$

where λ_{disc} is a weight applied to the discriminator loss that balances the detection loss.

2.2. Progressive Adaptation

Aligning feature distributions between two distant domains is challenging, and hence we introduce an intermediate feature space to make the adaptation task easier. That is, without directly solving the gap between the source and the target domains, we progressively perform adaptation to the target domain bridged by the intermediate domain.

Intermediate Domain. The intermediate domain is constructed from the source domain images to synthesize the target distributions on the pixel-level. We apply an image-to-image translation network, CycleGAN [24], to learn a function that maps the source domain images to the target ones, and vice versa. Since groundtruth labels are only available in the source domain, we only consider the translation from source images to the target domain (i.e., synthetic target images) after CycleGAN training.

Adaptation Process. Domain adaptation involves obtaining knowledge (feature) from a labeled source domain $\mathbb{S}$ and then apply that to an unlabeled target domain $\mathbb{T}$ by aligning the two distributions, solving the adaptation task $\mathbb{S} \rightarrow \mathbb{T}$. To take advantage of the intermediate domain during alignment, the proposed algorithm takes incremental steps and decomposes the problem into two stages: $\mathbb{S} \rightarrow \mathbb{F}$ and $\mathbb{F} \rightarrow \mathbb{T}$, as shown in Figure 2 a) and b). At stage one, we use $\mathbb{S}$ as the source domain adapting to $\mathbb{F}$ as the target domain without labels. Due to the underlying similarity between $\mathbb{S}$ and $\mathbb{F}$ in image contents, the network focuses on aligning the feature distributions with respect to the appearance difference in the pixel-level. After aligning the pixel discrepancies between $\mathbb{S}$ and $\mathbb{F}$, we take $\mathbb{F}$ as the source domain for supervision and adapts to $\mathbb{T}$ as stage two in the proposed method. During this step, the model takes advantage of the appearance invariant features from the previous step and focus on adapting the objects and context distributions. In summary, the proposed method separates the adaptation task into two subtasks and pays more attention to individual discrepancies during each adaptation stage.

Weighted Supervision. We observe that the quality of synthetic images differs in a wide range. For instance, some images fail to preserve details of objects or contain artifacts when translated, and these failure cases may have a larger distance to the target distribution.

As a result, when performing supervised detection learning on $\mathbb{F}$ during $\mathbb{F} \rightarrow \mathbb{T}$, these defects may cause confusions to our detection model, leading to false feature alignment across domains. To alleviate this problem, we propose an importance weighting strategy for synthetic samples based on their distances to the target distribution. Specifically, synthetic outliers that are farther away from the target distributions will receive less attention than the ones that are closer to the target domain. We obtain the weights by taking the predicted output scores from the target domain discriminator D_{cycle} in CycleGAN [24]. This discriminator is trained to differentiate between the source and target images with respect to the target distribution, in which the optimal discriminator is obtained with:

$$D^*_{cycle}(\mathbf{I}) = \frac{p_{\mathbb{T}}(\mathbf{I})}{p_{\mathbb{S}}(\mathbf{I}) + p_{\mathbb{T}}(\mathbf{I})}, \quad (4)$$

where $\mathbf{I}$ is the synthetic target image generated via CycleGAN, and $p_{\mathbb{T}}(\mathbf{I})$ and $p_{\mathbb{S}}(\mathbf{I})$ are the probability of $\mathbf{I}$ belonging to the source and the target domain, respectively. Here, the higher score of $D_{cycle}(\mathbf{I})$ represents a closer distribution to the target domain and thus should provide a higher weight. On the other hand, lower quality images which are further away from the target domain will be treated as outliers and receive a lower weight. For each image $\mathbf{I}$, the importance weight is defined as:

$$w(\mathbf{I}) = \begin{cases} D_{cycle}(\mathbf{I}), & \text{if } \mathbf{I} \in \mathbb{F} \\ 1, & \text{otherwise.} \end{cases} \quad (5)$$

We then apply this weight to the detection loss functions in equation (1) when learning from the synthetic image annotations during the second stage. Thus, the final weighted objective function given images $\mathbf{I}_{\mathbb{S}}$ and $\mathbf{I}_{\mathbb{T}}$ is re-written from (2) as:

$$\min_E \max_D \mathcal{L}(\mathbf{I}_{\mathbb{S}}, \mathbf{I}_{\mathbb{T}}) = w(\mathbf{I}_{\mathbb{S}})\mathcal{L}_{det}(\mathbf{I}_{\mathbb{S}}) + \lambda_{disc}\big[\mathcal{L}_{disc}(E(\mathbf{I}_{\mathbb{S}})) + \mathcal{L}_{disc}(E(\mathbf{I}_{\mathbb{T}}))\big]. \quad (6)$$

3. Experimental Results

In this section, we validate our method by evaluating the performance in two real-world scenarios that result in different domain discrepancies: *cross-camera adaptation*, and *weather adaptation*.

Table 1. Weather adaptation focusing on clear weather to foggy weather using the Cityscapes and Foggy Cityscapes datasets respectively. Performance is evaluated using the mean average precision (mAP) across 8 classes.

Cityscapes → Foggy Cityscapes									
Method	person	rider	car	truck	bus	train	motorcycle	bicycle	mAP
Faster R-CNN	23.3	29.4	36.9	7.1	17.9	2.4	13.9	25.7	19.6
FRCNN in the wild [2]	25.0	31.0	40.5	22.1	35.3	20.2	20.0	27.1	27.6
Ours (w/o synthetic)	30.2	37.9	46.1	14.7	26.9	7.0	20.8	31.5	26.9
Ours (synthetic augment)	**36.6**	45.3	**55.0**	24.2	43.9	18.5	28.4	**37.1**	36.1
Ours (progressive)	36.0	**45.5**	54.4	**24.3**	**44.1**	**25.8**	**29.1**	35.9	**36.9**
Oracle	37.8	48.4	58.8	25.2	53.3	15.8	35.4	39.0	39.2

Table 2. Cross camera adaptation using KITTI and Cityscapes datasets. The results show the average precision (AP) of the *car* class shared between the two domains.

KITTI → Cityscapes	
Method	AP
Faster R-CNN	28.8
FRCNN in the wild [2]	38.5
Ours (w/o synthetic)	38.2
Ours (synthetic augment)	40.6
Ours (progressive)	**43.9**
Oracle	55.8

3.1. Cross Camera Adaptation

The underlying camera settings and mechanisms of different datatsets can lead to critical differences in visual appearance as well as the image quality. These discrepancies are where the domain-shift takes place. In this experiment, we show the adaptation between images taken from different cameras and with distinctive scenery plus content differences. The KITTI [6] and Cityscapes [3] datasets are used as source and target respectively to conduct the cross camera adaptation experiment. Experimental results show that our method performs favorably against the state-of-the-art method [2] that learns to adapt in the feature space, matching the baseline performance of our own implementation denoted as Ours (w/o synthetic). The adaptation results are shown in Table 2, evaluated on the *car* class in terms of the average precision (AP).

In order to validate our method, we also conduct ablation studies using several settings. First, we demonstrate the benefit of utilizing information from the synthetic domain. When we directly augment synthetic data in the training set and include them in the source domain to perform feature-level adaptation, there is a 2.4% performance gain. In the proposed method, by adopting our progressive training scheme with the importance weights, we show in Table 2 that our model further improves the AP by 3.3%. Overall, our model can address the domain-shift problem caused by the camera along with other content differences across two distinct datasets and achieves state-of-the-art performance.

3.2. Weather Adaptation

Under real-world scenarios, object detection models can be applied in different weather conditions where they may not have sufficient knowledge of. However, it is difficult to obtain a large number of annotations in every weather condition for the models to learn. This section studies the weather adaptation from clear weather to a foggy environment. The Cityscapes [3] dataset is used as the source domain and the Foggy Cityscapes [19] dataset as the target domain.

For a fair comparison with the state-of-the-art method [2], we evaluate our method on 8 classes in the Cityscapes dataset as shown in Table 1. This table shows that our method can further reduce the domain gap across weather conditions. In addition, we discuss the characteristics of the two datasets and why it is in favor of our method.

When synthetic images are used during adaptation, the results show that there is a 9.2% improvement in performance. We note that the target Foggy Cityscapes dataset is fundamentally the same image as the source, Cityscapes dataset. On the other hand, the synthetic data is distributed closely to the target domain and inherits informative labels for the network to learn, enhancing performance in the target domain. Given such information learned from the synthetic domain, both our method and the synthetic augmented one climbs closely to the oracle result. Although the synthetic domain lies close to the target distribution, we show in the results that our progressive training can still assist the adaptation process, improving performance and at the same time generalizing well to different categories.

4. Conclusions

In this paper, we propose a progressive adaptation method that bridges the domain gap using an intermediate domain, decomposing a more difficult task into two easier subtasks. Using this intermediate domain, the proposed method progressively solves the adaptation subtasks by first adapting from source to the intermediate domain and then finally to the target domain. Experimental results show that the proposed method performs favorably against the state-of-the-art method and can further reduce the domain discrepancy under various scenarios.

References

[1] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan. Domain separation networks. In *NIPS*, 2016. 1

[2] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. V. Gool. Domain adaptive faster r-cnn for object detection in the wild. In *CVPR*, 2018. 1, 4

[3] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 4

[4] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, 2015. 3

[5] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 2016. 1

[6] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 4

[7] R. Girshick. Fast r-cnn. In *ICCV*, 2015. 1

[8] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *ICCV*, 2011. 1

[9] J. Hoffman, S. Guadarrama, E. S. Tzeng, R. Hu, J. Donahue, R. Girshick, T. Darrell, and K. Saenko. Lsda: Large scale detection through adaptation. In *NIPS*, 2014. 1

[10] J. Hoffman, D. Wang, F. Yu, and T. Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *CoRR*, abs/1612.02649, 2016. 1

[11] N. Inoue, R. Furuta, T. Yamasaki, and K. Aizawa. Cross-domain weakly-supervised object detection through progressive domain adaptation. In *CVPR*, 2018. 1

[12] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. In *ECCV*, 2016. 1

[13] M. Long, Y. Cao, J. Wang, and M. Jordan. Learning transferable features with deep adaptation networks. In *ICML*, 2015. 1

[14] M. Long, J. Wang, and M. I. Jordan. Deep transfer learning with joint adaptation networks. In *ICML*, 2017. 1

[15] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. *CVPR*, 2016. 1

[16] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. *CVPR*, 2017. 1

[17] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018. 1

[18] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *TPAMI*, 2017. 1, 2

[19] C. Sakaridis, D. Dai, and L. V. Gool. Semantic foggy scene understanding with synthetic data. *IJCV*, 2018. 4

[20] B. Sun and K. Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV Workshops*, 2016. 1

[21] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker. Learning to adapt structured output space for semantic segmentation. In *CVPR*, 2018. 1

[22] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *CVPR*, 2017. 1

[23] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. Deep domain confusion: Maximizing for domain invariance. *CoRR*, abs/1412.3474, 2014. 1

[24] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. 2, 3