SSF-DAN: Separated Semantic Feature based Domain Adaptation Network for Semantic Segmentation

Liang Du*[1], Jingang Tan[1], Hongye Yang[1], Jianfeng Feng[2], Xiangyang Xue[3], Qibao Zheng[2], Xiaoqing Ye[4] and Xiaolin Zhang[1,5]

[1]Bionic Vision System Laboratory, State Key Laboratory of Transducer Technology, Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai, China.
[2]Institute of Science and Technology for Brain-Inspired Intelligence, Fudan University, China.
[3]School of Computer Science, Fudan University, China.
[4]Baidu Inc.
[5]ShanghaiTech University, Shanghai, China.

Abstract

Despite the great success achieved by supervised fully convolutional models in semantic segmentation, training the models requires a large amount of labor-intensive work to generate pixel-level annotations. Recent works exploit synthetic data to train the model for semantic segmentation, but the domain adaptation between real and synthetic images remains a challenging problem. In this work, we propose a Separated Semantic Feature based domain adaptation network, named SSF-DAN, for semantic segmentation. First, a Semantic-wise Separable Discriminator (SS-D) is designed to independently adapt semantic features across the target and source domains, which addresses the inconsistent adaptation issue in the class-wise adversarial learning. In SS-D, a progressive confidence strategy is included to achieve a more reliable separation. Then, an efficient Class-wise Adversarial loss Reweighting module (CA-R) is introduced to balance the class-wise adversarial learning process, which leads the generator to focus more on poorly adapted classes. The presented framework demonstrates robust performance, superior to state-of-the-art methods on benchmark datasets.

1. Introduction

Semantic segmentation has been extensively studied due to its potential utilisation in autonomous driving [10, 40] and medical image processing [29]. A substantial number of works based on convolutional neural networks [3, 4, 20–

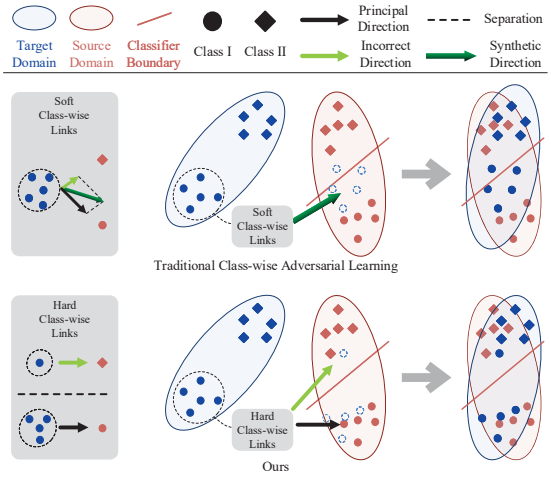

Figure 1: Illustration of the traditional class-wise adversarial learning method [6] and ours. Traditional class-wise adaptation considers all possible class-wise adaptation directions for the features. For unsupervised learning, it is common that some of these directions are incorrect. Consequently, these incorrect directions will influence the entire class-wise adaptation direction. In our method, for each feature, we take the class with highest proportion in its respective field of the prediction as principal adaptation direction to separate features for independent adaptation operation. It prevents the principal adaptation from being affected by the incorrect adaptation, which keeps the clear classifier boundary among the adapted features.

*duliang@mail.ustc.edu.cn

22, 41, 44, 45] have been introduced to address the problem with pixel-wise annotated images. However, building large-scale and per-pixel labelled datasets for a semantic segmentation task would require intensive human labor with adequate expertise. Therefore, exploring economical approaches for acquiring semantic segmentation-specific data is appealing, i.e., using synthetic data [11, 28], by which pixel-level annotations can automatically be generated at a lower cost. However, synthetic data still suffer from a substantial domain difference from real-world data, which results in a dramatic performance drop when applying the model to real-world scenes. In light of this issue, domain adaptation techniques have been proposed to close the gap between the target domain and source domain. Previous works on domain adaptation techniques [18, 42] attempted to either minimize the difference between the source and target feature distributions, or explicitly enforced two data distributions to be close to each other through adversarial learning. For image-level classification tasks, features are aligned across the source and target domains based on a generative adversarial network [9, 23] such that the adapted features are able to generalize in both domains. However, for pixel-level classification tasks such as semantic segmentation, the network needs to extract and encode various visual features for diverse semantic objects. As mentioned in [19], a whole-image discriminator for validating the fidelity of all regions makes the color/texture of all pixels in original images easily collapse into a monotonous pattern, which would severely hinder the capabilities of facilitating downstream vision perception tasks. We also believe that the feature distributions for each class should be regarded differently to take the semantic consistency into consideration in adversarial learning. [6] introduced a joint global and class-wise adversarial learning framework, but the result was affected by the inconsistent adaptation as depicted in Figure 1, which will be further explained in Section 3.3. Output-space-based adversarial learning has been proposed by [34] and achieved great success. However, it does not fully utilize high-dimensional features.

In this paper, we introduce an unsupervised domain adaptation network via class-wise adversarial learning for semantic segmentation. We introduce SS-D to evaluate the feature alignment quality in an independent semantic-wise manner to bridge the domain gap in each class without causing the inconsistent adaptation. The segmentation model and our SS-D are jointly trained in an end-to-end manner, and no prior knowledge of target domain data is utilized. SS-D is directly discarded during the test phase.

Our main contributions can be summarized as follows:

- We propose a novel end-to-end framework for semantic segmentation via independent class-wise adversarial learning without any global feature alignment.
- We propose a Semantic-wise Separable Discriminator (SS-D) to independently adapt separated semantic features from the target to source domain with progressive confidence strategy to address the critical inconsistent adaptation issue in the class-wise adversarial learning.
- We propose a Class-wise Adversarial loss Reweighting module (CA-R) to enforce the generator to pay more attention to those weakly adapted classes.
- Our framework achieves new state-of-the-art performance on benchmark datasets.

2. Related Work

Semantic Segmentation Due to recent advances in fully convolutional networks (FCNs), semantic segmentation has received increasing attention from both academia and industry. Using FCN for pixel-level classification was first introduced by Long et al. [22]. Numerous approaches have subsequently been explored to improve this model. Dilated convolutions were used in [2, 41] to enlarge the receptive field of neural networks. A pyramid pooling module was recently presented in [44] to encode the global and local context. However, these cutting edge approaches rely on a significant amount of pixel-level annotated data. Synthetic datasets based on rendering (e.g., GTA5 [28] and Synthia [30]) are constructed to alleviate the annotation issue, since their pixel-level labels can be generated with a partially automated process. Nevertheless, due to discrepancies [26] in data distributions, a synthetic dataset cannot be directly used to train the model for real-world applications. Consequently, domain adaptation techniques are appealing to be developed.

Domain adaptation For vision tasks such as image classification, domain adaptation approaches have been proposed to narrow the domain gap between the source and target data. By aligning the feature distributions between the source and target images [5, 8, 9, 16, 23–25, 31, 32, 35, 36, 38, 47], the generalization ability of the model will be improved. The domain adversarial neural network (DANN) was first introduced by Ganin et al. [8, 9] to transfer the feature distribution. For pixel-level classification, [15] was the first to apply adversarial learning in a fully convolutional way to perform feature adaptation. [37] addressed the task of unsupervised domain adaptation in semantic segmentation with losses based on the entropy of the pixel-wise predictions. Another approach to solve the domain adaptation issue is to apply a style transfer technique to stylize annotated source domain images as target domain images. [19] introduced a Semantic-aware Grad-GAN to transfer personalized styles for distinct semantic regions in synthetic images to approximate the real-world distributions based on ground

truth semantic labels. [27] elaborated a cycle-consistent adaptation framework based on the style transfer network Cycle-GAN [46], which combined the cycle-consistent loss with adversarial loss to minimize both feature and pixel-level domain gaps. Other methods [6, 15] have focused on adapting synthetic-to-real or cross-city images by adopting class-wise adversarial learning. [6] proposed a global and class-wise adversarial learning framework to adapt road scene segmenters across diverse cities. Self-training [1, 48] is an alternative way to perform domain adaptation for many vision tasks [33, 49]. [49] introduced a CNN-based self-training framework for domain adaptation in semantic segmentation that unifies the feature space alignment and the task itself together under a single, unified loss.

3. Method

3.1. Overview of the Framework

The overall framework is depicted in Figure 2. Our SSF-DAN consists of three major components: the segmenter (generator), including G and σ, which transforms the input image to a high-level feature space and maps the feature space to the output label space; the SS-D (discriminator) D for independent class-wise alignment; and the CA-R module for reweighting the class-wise adversarial loss.

The image I_s from the source domain is first passed to the segmenter with its annotation Y_s to optimize the generator. Next, the network predicts the semantic segmentation output P_t for the image I_t from the target domain. We filter out pixels with low confidences in the pseudo label P_t by our progressive confidence strategy, which is explained in Section 3.3. Then, we separate the semantic features from the last feature by the downsampled one-hot output (according to the mode). Subsequently, we forward these feature blocks to the corresponding convolutional layers of the discriminator to distinguish whether the input class features are from the source or target domain. An adversarial loss on the target prediction makes the network propagate gradients from D to G and enforces G to generate feature distributions similar to those of the source domain. Note that the feature blocks for related convolutional layers of SS-D are semantic-wisely separated to guarantee the independent adaptation, which is further explained in Section 3.3. Finally, the CA-R module calculates the class-wise weight maps R_t and R_s according to P_t and reweights the class-wise adversarial loss, which is detailed in Section 3.4.

3.2. Objective Function for Domain Adaptation

To close the domain gap between the source and target data and perform the segmentation task, the objective function is defined as follows:

$$\mathcal{L}(I_s, Y_s, I_t) = \mathcal{L}_{seg}(I_s, Y_s) + \delta \mathcal{L}_{adv}(I_s, I_t) \qquad (1)$$

where $\mathcal{L}_{seg}$ is the cross-entropy loss of the segmentation in the source domain; $\mathcal{L}_{adv}$ is the adversarial loss, which minimizes the gap between the source and target domains; and δ is the balanced weight of $\mathcal{L}_{seg}$ and $\mathcal{L}_{adv}$.

3.3. Semantic-wise Separable Discriminator

Although the idea of performing class-wise adaptation has been proposed in [6], we argue that this kind of class-wise alignment is inconsistent because their multiple class-wise discriminators are not independent in terms of the responding area in the last feature, which severely limits the potential capability of class-wise adversarial learning.

[6] introduces the "soft" class-wise weight map W^c_{soft} and takes each grid in it as an instance. c represents the class. The grid is calculated based on the class proportion among all pixels in its respective field on the output predicted label. W^c_{soft} is multiplied to the output of each discriminator to calculate the class-wise adversarial loss. Each discriminator individually focuses on the distinct semantic regions of the entire feature, according to the receptive fields of all nonzero pixels in its related W^c_{soft}. It is common that there are overlaps in these regions of different discriminators.

For the features extracted from one class object, these overlaps are caused by incorrect nonzero predictions. Consequently, such features may respond to multiple discriminators and be inconsistently adapted. Specifically, the gradients of the weights related to the true object class k in the generator is defined as follows:

$$\nabla_{\mathcal{W}_G^{\gamma(k)}} = \frac{\partial L_k}{\partial \mathcal{W}_G^{\gamma(k)}} + \sum_{i \in C, i \neq k} \frac{\partial L_i}{\partial \mathcal{W}_G^{\gamma(k)}} \qquad (2)$$

where $\mathcal{W}_G^{\gamma(k)}$ is the set of weights responding to class k in the generator and $\gamma(k)$ represents the set related to class k. L_i is the loss of the class i discriminator. C represents all classes. For independent class-wise adaptation, $\nabla_{\mathcal{W}_G^{\gamma(k)}}$ should be equal to the first part in (2), while the second part introduced by W^c_{soft} in [6] is a noise term, which will result in the inconsistent adaptation. For some features extracted from the boundary regions with more than one class object, W^c_{soft} will enforce the generator to adapt the features to diverse feature spaces. Nevertheless, it is difficult for the generator to adapt such features to corresponding multiple feature spaces simultaneously, which may also cause the inconsistent adaptation or destroy the existing alignment.

In supervised learning, it is better to utilize the "soft" label which obtains more information for training the model. However, for unsupervised domain adaptation, the reliability of the information can not be guaranteed. Even if some of the information is sufficiently reliable, the generator will still lack the adversarial capability to tackle the multiple discriminators simultaneously. Therefore, the key point in the unsupervised class-wise adversarial learning is to make

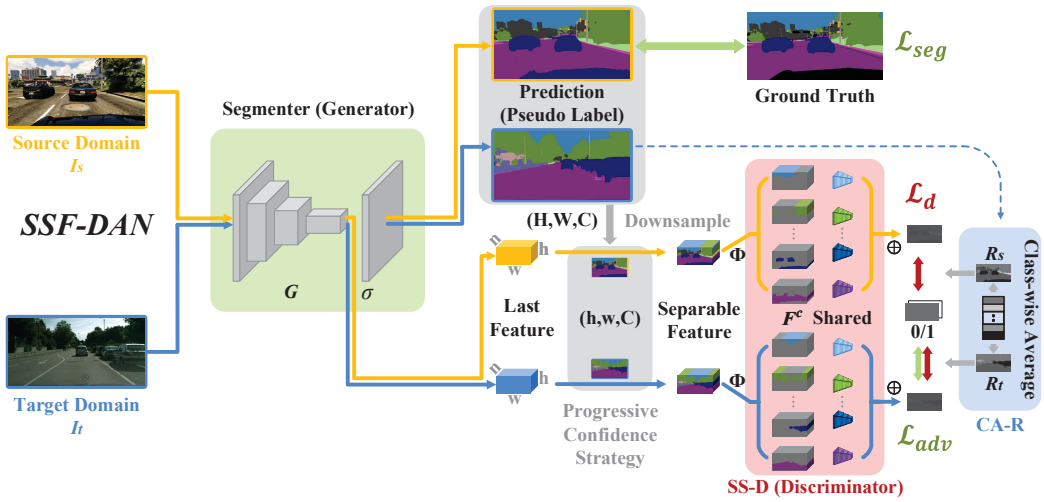

Figure 2: The overview of SSF-DAN. Images in the source and target domains are randomly selected and passed through the generator to get output predictions. For the source prediction, a segmentation loss is computed based on the source ground truth. We separate semantic features from the last feature according to the downsampled pseudo label and pass them to specific convolutional layers in our SS-D. Then SS-D distinguishes whether the class-wise features are from the source or target domain. An adversarial loss is calculated on the target prediction and is back-propagated to the segmenter. CA-R module is applied to reweight the class-wise adversarial loss based on the target prediction. The progressive confidence strategy is used for more reliable pseudo label. Φ and $\oplus$ represent the semantic-wise separation and the channel-wise summation operation.

the class-wise adaptation process independent to prevent it from being affected by ambiguous information.

Compared with state-of-the-art class-wise adaptation approaches [6, 15], our improvements are as follows: (a) We separate different semantic features from the entire feature space according to the downsampled pseudo label to make the class-wise adaptation independent. As depicted in Figure 1, most of the features will be adapted to their principal class-wise feature space without being affected by the incorrect information. (b) The progressive confidence strategy will also decrease the incorrect adaptation during the adaptation process. Note that our method assumes that target samples with higher prediction probability have better prediction accuracy [49].

Class-wise adversarial learning. Figure 2 presents an illustration of the proposed SS-D. The segmentation cross-entropy loss defined in (1) is formulated as follows:

$$\mathcal{L}_{seg}(I_s, Y_s) = -\sum_{H,W}\sum_{c\in\mathcal{C}} Y_s^{(H,W,C)} log(P_s^{(H,W,C)}) \tag{3}$$

where Y_s is the ground-truth annotations for images from the source domain; $P_s = \sigma(F_s) = \sigma(G(I_s))$ is the segmentation output; F_s is the last feature map; and σ is the decoder including the convolution, upsample and softmax operations. After forwarding the source domain image and calculating the segmentation loss, the target domain images are forwarded to G, and the prediction is $P_t = \sigma(F_t) = \sigma(G(I_t))$. We denote M_t and M_s as the final one-hot outputs of P_t and P_s and separate the class masks M_t^c and M_s^c from M_t and M_s by the class channel. Note that $P, Y, M \in \mathbb{R}^{H\times W\times C}$ and $F \in \mathbb{R}^{h\times w\times n}$, where n represents the feature channels. F_t is multiplied by the downsampled M_t^c channel-wise to obtain the semantic feature block F_t^c. In other words, we preserve the value in the region of interest and set the other region to zero for each semantic feature block. Then, each feature block is forwarded to the related convolutional layers of our SS-D. Finally, all class outputs are summed into a single-channel output, and the output is compared with an all-zero tensor to calculate $\mathcal{L}_{adv}$ in (1), which is defined as follows:

$$\mathcal{L}_{adv} = -\sum_{h,w}\sum_{c\in\mathcal{C}} log(1 - D^c(F_t^c)^{(h,w,1)}) \tag{4}$$

where D^c represents the specific convolution operation of class c in our SS-D. This adversarial loss is optimized for cheating the discriminator by maximizing the probability of the target prediction being considered as the source one.

After the generation process, the generator's parameters are frozen. We forward F_t^c and F_s^c to our SS-D using a cross-entropy loss $\mathcal{L}_d$ for the source and target classes. The loss function is defined as follows:

$$\mathcal{L}_d = -\sum_{h,w}\sum_{c\in\mathcal{C}} [(1-\alpha) log(1 - D^c(F^c)^{(h,w,1)}) + \alpha log(D^c(F^c)^{(h,w,1)})] \tag{5}$$

where $\alpha = 0$ if the sample is drawn from the source domain and $\alpha = 1$ for a sample from the target domain. We optimize the following min-max criterion:

$$\max_{D} \min_{G} \mathcal{L}(I_s, Y_s, I_t) \tag{6}$$

The goal of our framework is to minimize the segmentation loss in G for source images while maximizing the probability of target predictions being considered as source ones.

Progressive confidence strategy for the more reliable pseudo label. At the beginning of training, the confidences of the pseudo labels in each class are low and not sufficiently reliable for training. The confidences increase gradually with the training process, and more reliable pixels in the label can be used. To maintain the reliability of the pseudo label during the whole training process, we set a hyperparameter ρ to control the proportion of pixels preserved in the pseudo label. Specifically, M is multiplied by P in an element-wise manner, and the result can be separated by the class channel to obtain the class-wise confidence map A^c. In each A^c, all the confidence values are sorted in descending order. Thus, we can filter out unreliable predicted pixels with low confidences in each class according to ρ. ρ is varying progressively from low to high during the training process:

$$\rho = \begin{cases} \epsilon/\varepsilon & if \quad \epsilon/\varepsilon < \rho_{upper} \\ \rho_{upper} & else \end{cases} \tag{7}$$

where ε is the total iteration step and ϵ is the current one. ρ_{upper} is the upper bound of the preserved labels' proportion. Experiments show that $\rho_{upper} = 0.8$ performs the best. The comparison of different ρ_{upper} is shown in Figure 4. The more reliable class-wise confidence map A^{c*} and mask M^{c*} are obtained as follows:

$$\begin{aligned} A^{c*} &= \mu(A^c, \rho) \\ M^{c*} &= \mu(M^c, \rho) \end{aligned} \tag{8}$$

where $\mu(\cdot, \rho)$ is the operation to filter out low-confidence pixels by the progressive strategy with current ρ. The loss from the filtered pixels is removed by M^{c*}.

3.4. Class-wise Adversarial Loss Reweighting

The parameters of different convolutional layers for the discriminator are updated independently, whereas the parameters of the generator are not. Although the capability of the discriminator is improved due to our class-wise adaptation method, it is much more difficult for the generator to fool the better discriminator. Since we assume that target samples with higher prediction probability have better prediction accuracy, those classes with higher prediction probability are well adapted and predicted. To make the generator focus on poorly adapted classes to balance the class-wise adversarial learning process, we propose the CA-R module to adaptively reweight the adversarial learning loss based on A_t^c. The class-wise reweight value τ_t^c is defined as follows:

$$\tau_t^c = \sqrt{N^c / \sum_{i=0}^{N^c} A_t^{c*}} \tag{9}$$

N^c is the nonzero pixel number of A_t^{c*}, and i is the index of these pixels. First we use τ_t^c to replace the nonzero values of the related class masks M_t^{c*} and M_s^{c*}. Then, we merge the results into one channel and downsample them to match the discriminator's output size to obtain the reweight maps R_t and R_s, as shown in Figure 2. R_t and R_s serve as the weights of the adversarial loss in the target and source domains. Our CA-R decreases the weights of the loss for those classes with higher average confidence, so the generator will pay more attention to poorly adapted classes.

3.5. Network Architecture

Segmentation Network Inspired by [34], we adopt the DeepLab-v2 [3] framework with ResNet-101 [13] model pretrained on ImageNet [7] as our segmentation baseline network. Similar to [3, 41], the last classification layer is removed, and the strides of the last two convolution layers are modified from 2 to 1, which makes the resolution of the output feature maps one-eighth of the input image size. Dilated convolution layers are applied [41] in conv4 and conv5 layers with strides of 2 and 4, respectively, to enlarge the receptive field. After the last layer, atrous spatial pyramid pooling (ASPP) [3] is utilized as the final classifier. Finally, an upsampling layer is applied along with the softmax output to match the size of the input image.

Discriminator Our SS-D uses an FCN architecture. Each class-wise convolution module consists of 5 convolutional layers. The kernel sizes are 1, 3, 3, 3, 3 and the numbers of channels are 2048, 1024, 512, 256, 128, 1. Each convolutional layer is followed by a leaky ReLU [12] parameterized by 0.2, except for the last layer.

4. Experiments

4.1. Datasets and Experimental Setup

Datasets Our segmentation network is trained on two source datasets, GTA5 and Synthia, and the models are evaluated on the target dataset, Cityscapes. Experiments are also performed on the Cross-City dataset [6]. We train the model on one city (Cityscapes) with supervision and adapt the model to another city without any supervision.

Cityscapes is a real-world dataset that contains street scenes of 50 different cities, totalling 5000 pixel-level-annotated images. The dataset is divided into a training set with 2993 images, a validation set with 503 images and a

Table 1: Results of adapting GTA5 to Cityscapes. We compare our results with those of state-of-the-art approaches with VGG-16 and DeepLab-V2 based models. The first row of each method (source only) represents the model without adaptation.

Method	Base Net	road	sidewalk	building	wall	fence	pole	light	sign	veg	terrain	sky	person	rider	car	truck	bus	train	mbike	bike	mIoU
Source only	Dilation-Frontend	31.9	18.9	47.7	7.4	3.1	16.0	10.4	1.0	76.5	13.0	58.9	36.0	1.0	67.1	9.5	3.7	0.0	0.0	0.0	21.2
FCNs Wild [15]	[41]	70.4	32.4	62.1	14.9	5.4	10.9	14.2	2.7	79.2	21.3	64.6	44.1	4.2	70.4	8.0	7.3	0.0	3.5	0.0	27.1
Source only	FCN8s-VGG16	18.1	6.8	64.1	7.3	8.7	21.0	14.9	16.8	45.9	2.4	64.4	41.6	17.5	55.3	8.4	5.0	6.9	4.3	13.8	22.3
Curr. DA [43]	[22]	74.9	22.0	71.7	6.0	11.9	8.4	16.3	11.1	75.7	13.3	66.5	38.0	9.3	55.2	18.8	18.9	0.0	16.8	16.6	28.9
Source only	FCN8s-VGG16	26.0	14.9	65.1	5.5	12.9	8.9	6.0	2.5	70.0	2.9	47.0	24.5	0.0	40.0	12.1	1.5	0.0	0.0	0.0	17.9
CyCADA [14]	[22]	85.2	37.2	76.5	21.8	15.0	23.8	22.9	21.5	80.5	31.3	60.7	50.5	9.0	76.9	17.1	28.2	4.5	9.8	0.0	35.4
Source only	DeepLab-v2	75.8	16.8	77.2	12.5	21.0	25.5	30.1	20.1	81.3	24.6	70.3	53.8	26.4	49.9	17.2	25.9	6.5	25.3	36.0	36.6
AdaptSegNet [34]	[17]	86.5	36.0	79.9	23.4	23.3	23.9	35.2	14.8	83.4	33.3	75.6	58.5	27.6	73.7	32.5	35.4	3.9	**30.1**	28.1	42.4
Source only CBST [49]	FCN8s-VGG16	64.0	22.1	68.6	13.3	8.7	19.9	15.5	5.9	74.9	13.4	37.0	37.7	10.3	48.2	6.1	1.2	1.8	10.8	2.9	24.3
	[22]	66.7	26.8	73.7	14.8	9.5	28.3	25.9	10.1	75.5	15.7	51.6	47.2	6.2	71.9	3.7	2.2	5.4	18.9	32.4	30.9
	ResNet-38	70.0	23.7	67.8	15.4	18.1	40.2	41.9	25.3	78.8	11.7	31.4	62.9	29.8	60.1	21.5	26.8	7.7	28.1	12.0	35.4
	[39]	86.8	**46.7**	76.9	26.3	**24.8**	**42.0**	**46.0**	**38.6**	80.7	15.7	48.0	57.3	27.9	78.2	24.5	**49.6**	**17.7**	25.5	**45.1**	45.2
Source only Ours	FCN8s-VGG16	64.0	22.1	68.6	13.3	8.7	19.9	15.5	5.9	74.9	13.4	37.0	37.7	10.3	48.2	6.1	1.2	1.8	10.8	2.9	24.3
	[22]	88.7	32.1	79.5	**29.9**	22.0	23.8	21.7	10.7	80.8	29.8	72.5	49.5	16.1	82.1	23.2	18.1	3.5	24.4	8.1	37.7
	DeepLab-v2	75.8	16.8	77.2	12.5	21.0	25.5	30.1	20.1	81.3	24.6	70.3	53.8	26.4	49.9	17.2	25.9	6.5	25.3	36.0	36.6
	[17]	**90.3**	38.9	**81.7**	24.8	22.9	30.5	37.0	21.2	**84.8**	**38.8**	**76.9**	**58.8**	**30.7**	**85.7**	30.6	38.1	5.9	28.3	36.9	**45.4**

Table 2: Results of adapting Synthia to Cityscapes.

Method	Base Net	road	sidewalk	building	light	sign	veg	sky	person	rider	car	bus	mbike	bike	mIoU
Source only	Dilation-Frontend	6.4	17.7	29.7	0.0	7.2	30.3	66.8	51.1	1.5	47.3	3.9	0.1	0.0	20.2
FCNs Wild [15]	[41]	11.5	19.6	30.8	0.1	11.7	42.3	68.7	51.2	3.8	54.0	3.2	0.2	0.6	22.1
Source only	FCN8s-VGG16	5.6	11.2	59.6	8.0	5.3	72.4	75.6	35.1	9.0	23.6	4.5	0.5	18.0	27.6
Curr. DA [43]	[22]	65.2	26.1	74.9	3.5	3.0	76.1	70.6	47.1	8.2	43.2	20.7	0.7	13.1	34.8
Source only	DeepLab-v2	55.6	23.8	74.6	6.1	12.1	74.8	79.0	55.3	19.1	39.6	23.3	13.7	25.0	38.6
AdaptSegNet [34]	[17]	84.3	**42.7**	77.5	4.7	7.0	77.9	82.5	54.3	21.0	72.3	32.2	18.9	32.3	46.7
Source only CBST [49]	FCN8s-VGG16	17.2	19.7	47.3	3.0	9.1	71.8	78.3	37.6	4.7	42.2	9.0	0.1	0.9	26.2
	[22]	69.6	28.7	69.5	11.9	13.6	82.0	81.9	49.1	14.5	66.0	6.6	3.7	32.4	36.1
	ResNet-38	32.6	21.5	46.5	4.8	13.1	70.8	60.3	56.6	3.5	74.1	20.4	8.9	13.1	33.6
	[39]	53.6	23.7	75.0	**23.5**	**26.3**	**84.8**	74.7	**67.2**	17.5	**84.5**	28.4	15.2	**55.8**	48.4
Source only Ours	FCN8s-VGG16	17.2	19.7	47.3	3.0	9.1	71.8	78.3	37.6	4.7	42.2	9.0	0.1	0.9	26.2
	[22]	**87.1**	36.5	79.7	13.5	7.8	81.2	76.7	50.1	12.7	78.0	35.0	4.6	1.6	43.4
	DeepLab-v2	55.6	23.8	74.6	6.1	12.1	74.8	79.0	55.3	19.1	39.6	23.3	13.7	25.0	38.6
	[17]	84.6	41.7	**80.8**	11.5	14.7	80.8	**85.3**	57.5	**21.6**	82.0	**36.0**	**19.3**	34.5	**50.0**

test set with 1531 images as well as 20021 auxiliary images. 34 distinct categories are contained in the dataset.

GTA5 is rendered from a computer game (Grand Theft Auto V): It contains 24966 high-resolution images, automatically annotated into 19 classes. The annotations are fully compatible with those of Cityscapes; thus, all 19 of the official training classes are used in our experiments.

Synthia is a large-scale synthetic dataset automatically generated for the semantic segmentation of urban scenes. As in [10, 15], we utilize Synthia-Rand-Cityscapes, a subset that contains 9400 images paired with Cityscapes, sharing 12 common classes, one void class and some unnamed classes. The synthetic images do not correspond to any of the real cities covered by Cityscapes.

NTHU is a real-world dataset with small domain gaps between cities. The dataset contains 4 different cities, and there are 100 image-annotation pairs for 13 classes shared with Cityscapes in each city and 3200 images without annotations. Following [6], we use the Cityscapes training set as the source domain and adapt the model to each target city in Cross-City using 3200 images without annotations. Another 100 annotated images are utilized for evaluation.

Experimental setup Following [34], we use the Cityscapes validation set as the test set. 500 validation images are randomly selected from the training set to monitor the convergence of the networks. During training, we randomly sample minibatches from the source images paired with their labels and the target images.

4.2. Implementation Details

For fair comparison with other methods, in addition to ResNet-101, we use FCN8s-VGG16 as our base network in GTAV to Cityscapes and Synthia to Cityscapes. In the cross-city setting, ResNet-101 is used as the base network to show the state-of-the-art performance. Our network is implemented using the PyTorch framework and tested on a P100 GPU with 16 GB of memory. For the semantic segmentation network, we use the stochastic gradient descent (SGD) optimizer with Nesterov acceleration. The momentum is 0.9, and the weight decay is 1e-4. The initial learning rate is set as $2.5 \times$ 1e-4 and is decreased using polynomial decay with a power of 0.9 as mentioned in [3]. For our SS-D, we use the Adam optimizer with the learning rate as 1e-4 and the same polynomial decay as the segmentation network. The momentum is set as 0.9. δ in (1) is set to

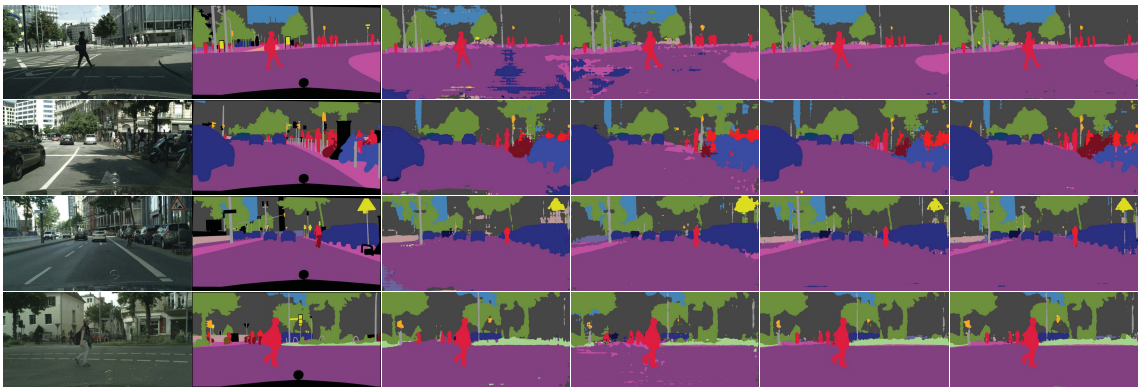

Figure 3: Example results of the adapted segmentation for GTA5-to-Cityscapes. For each Cityscapes image, we show the result before adaptation, the result based on the method used in [6] (Soft Class-wise + Global), the result based on our SS-D and on our full approach (SS-D + CA-R).

Table 3: Results of adapting Cityscapes to the Cross-City dataset. We apply the DeepLab-V2 architecture as in [34] and compare the results among state-of-the-art approaches. The first row of each method represents the model without adaptation.

City	Method	road	sidewalk	building	light	sign	veg	sky	person	rider	car	bus	mbike	bike	mIoU
Rome	Source Dilation-Frontend	77.7	21.9	83.5	0.1	10.7	78.9	88.1	21.6	10.0	67.2	30.4	6.1	0.6	38.2
	CrossCityAdapt [6]	79.5	29.3	84.5	0.0	22.2	80.6	82.8	29.5	13.0	71.7	37.5	25.9	1.0	42.9
	DeepLab-V2	83.9	34.3	87.7	13.0	41.9	84.6	92.5	37.7	22.4	80.8	38.1	39.1	5.3	50.9
	AdaptSegNet [34]	83.9	34.2	88.3	18.8	40.2	**86.2**	**93.1**	47.8	21.7	80.9	**47.8**	48.3	8.6	53.8
	Source Resnet-38	86.0	21.4	81.5	14.3	47.4	82.9	59.8	30.8	20.9	83.1	20.2	40.0	5.6	45.7
	CBST [49]	**87.1**	**43.9**	**89.7**	14.8	**47.7**	85.4	90.3	45.4	**26.6**	**85.4**	20.5	49.8	**10.3**	53.6
	DeepLab-V2	83.9	34.3	87.7	13.0	41.9	84.6	92.5	37.7	22.4	80.8	38.1	39.1	5.3	50.9
	Ours	84.2	38.4	87.4	**23.4**	43.0	85.6	88.2	**50.2**	23.7	80.6	38.1	**51.6**	8.6	**54.1**
Rio	Source Dilation-Frontend	69.0	31.8	77.0	4.7	3.7	71.8	80.8	38.2	8.0	61.2	38.9	11.5	3.4	38.5
	CrossCityAdapt [6]	74.2	43.9	79.0	2.4	7.5	77.8	69.5	39.3	10.3	67.9	41.2	27.9	10.9	42.5
	DeepLab-V2	76.6	47.3	82.5	12.6	22.5	77.9	86.5	43.0	19.8	74.5	**36.8**	29.4	16.7	48.2
	AdaptSegNet [34]	76.2	44.7	84.6	9.3	25.5	**81.8**	**87.3**	55.3	32.7	74.3	28.9	43.0	27.6	51.6
	Source Resnet-38	80.6	36.0	81.8	**21.0**	**33.1**	79.0	64.7	36.0	21.0	73.1	33.6	22.5	7.8	45.4
	CBST [49]	**84.3**	**55.2**	**85.4**	19.6	30.1	80.5	77.9	55.2	28.6	**79.7**	33.2	37.6	11.5	52.2
	DeepLab-V2	76.6	47.3	82.5	12.6	22.5	77.9	86.5	43.0	19.8	74.5	**36.8**	29.4	16.7	48.2
	Ours	74.2	43.7	82.5	10.3	21.7	79.4	86.7	**55.9**	**36.1**	74.9	33.7	**52.6**	**33.7**	**52.7**
Tokyo	Source Dilation-Frontend	81.2	26.7	71.7	8.7	5.6	73.2	75.7	39.3	14.9	57.6	19.0	1.6	33.8	39.2
	CrossCityAdapt [6]	83.4	**35.4**	72.8	12.3	12.7	77.4	64.3	42.7	21.5	64.1	20.8	8.9	40.3	42.8
	DeepLab-V2	82.9	31.3	78.7	14.2	24.5	81.6	89.2	48.6	33.3	70.5	**7.7**	11.5	45.9	47.7
	AdaptSegNet [34]	81.5	26.0	77.8	17.8	26.8	82.7	**90.9**	55.8	**38.0**	72.1	4.2	24.5	50.8	49.9
	Source Resnet-38	83.8	26.4	73.0	6.5	27.0	80.5	46.6	35.6	22.8	71.3	4.2	10.5	36.1	40.3
	CBST [49]	**85.2**	33.6	**80.4**	8.3	**31.1**	**83.9**	78.2	53.2	28.9	**72.7**	4.4	27.0	47.0	48.8
	DeepLab-V2	82.9	31.3	78.7	14.2	24.5	81.6	89.2	48.6	33.3	70.5	**7.7**	11.5	45.9	47.7
	Ours	82.1	27.4	78.0	**18.4**	26.6	83.0	90.8	**57.1**	35.8	72.0	4.6	**27.3**	**52.8**	**50.4**
Taipei	Source Dilation-Frontend	77.2	20.9	76.0	5.9	4.3	60.3	81.4	10.9	11.0	54.9	32.6	15.3	5.2	35.1
	CrossCityAdapt [6]	78.6	28.6	80.0	13.1	7.6	68.2	82.1	16.8	9.4	60.4	34.0	26.5	9.9	39.6
	DeepLab-V2	83.5	33.4	**86.6**	12.7	16.4	77.0	**92.1**	17.6	13.7	70.7	37.7	44.4	18.5	46.5
	AdaptSegNet [34]	81.7	29.5	85.2	**26.4**	15.6	76.7	91.7	31.0	12.5	71.5	**41.1**	47.3	27.7	49.1
	Source Resnet-38	84.9	26.0	80.1	8.3	**28.0**	73.9	54.4	18.9	26.8	71.6	26.0	48.2	14.7	43.2
	CBST [49]	**86.1**	35.2	84.2	15.0	22.2	75.6	74.9	22.7	**33.1**	**78.0**	37.6	**58.0**	**30.9**	50.3
	DeepLab-V2	83.5	33.4	**86.6**	12.7	16.4	77.0	**92.1**	17.6	13.7	70.7	37.7	44.4	18.5	46.5
	Ours	84.5	**35.3**	86.4	17.7	16.9	**77.7**	91.3	**31.8**	22.3	73.7	**41.1**	55.9	28.5	**51.0**

0.001 for GTA5, Synthia, and to 0.0005 for Cross-City.

4.3. Comparison with State-of-the-art Methods

Table 1 and 2 show the comparisons with state-of-the-art domain adaptation methods for semantic segmentation respectively on the GTA5 $\Rightarrow$ Cityscape and the Synthia $\Rightarrow$ Cityscape setting. The performance of our method is superior to the cutting edge adversarial learning methods in almost all classes, as clearly demonstrated in Table 1 and 2. Compared with the state-of-the-art self-training method

[49], although we are slightly weaker in some small object classes, "SSF-DAN" has a better overall performance. For those regions which are difficult to distinguish (e.g., the sky and building), "SSF-DAN" outperforms other methods by considerable margins. As shown in Table 3, for small domain adaptation across real-world cities, our method also outperforms state-of-the-art methods.

4.4. Ablation Studies

We investigate the effectiveness of the different modules of our method. We perform ablation experiments on the GTA5 $\Rightarrow$ Cityscape setting. We also utilize annotated ground truths in the Cityscapes dataset to train the model as the oracle results to measure how much the gap between the fully supervised model and the adapted model is narrowed.

Effectiveness of the SS-D. To verify the improvement of our class-wise adaptation, we perform a comparison with the state-of-the-art class-wise adaptation method [6]. The experimental results for different settings are shown in Table 4. The first row shows the result without adaptation. The second to fourth rows show the results with only global feature alignment, "soft" weight maps based class-wise alignment, and both as utilized in [6]. Our SS-D† (SS-D without progressive confidence strategy) outperforms "soft" class-wise (with global) alignment by 6.5%, which demonstrates that independent adaptation is much more important than adaptation considering of all possible classes in the unsupervised class-wise adaptation. To assess the impact of global feature adaptation on our class-wise adaptation, we further add the global alignment to our SS-D†. The results are slightly affected by the global alignment due to the introduction of inconsistent adaptation. The progressive confidence strategy contributes 2.8% mIoU gain in our SS-D. Note that we set $\rho_{upper} = 0.8$ in our progressive strategy. We also conducted the experiment which fixes $\rho = \rho_{upper}$ during the whole adaptation process to further verify the impact of our progressive strategy. These experimental comparisons under different ρ_{upper} settings are shown in Figure 4. More ablation experiments are detailed in our supplementary material.

Effectiveness of the CA-R module. As shown in Table 4, our CA-R module improves upon the results with only SS-D by 3.2%, which demonstrates that such adaptive class-wise balance strategy could improve the overall performance in the class-wise domain adaptation for semantic segmentation tasks.

Figure 3 presents some example results for the adapted segmentation. Substantial improvement is observed when we utilize our SS-D instead of the "soft" class-wise (with global) alignment method. We can see that each class region is clean and accurate due to our advanced method. The CA-R module further improves the accuracy of the classes that are often weakly adapted (e.g., sign and light) due to the balance of the class-wise adaptation process.

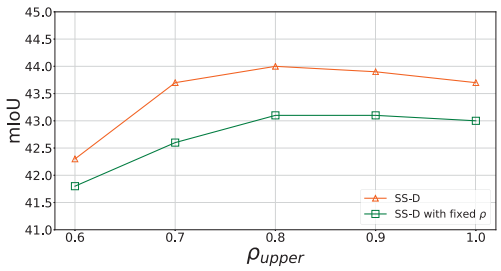

Figure 4: The comparison of using different ρ_{upper} in our SS-D. Results of adapting GTA5 to Cityscapes.

Finally, as depicted in Figure 3, our model still has limited capability to distinguish dense and tiny objects. This issue is worth considering and left for future work.

Table 4: Ablation study for different settings. Results of adapting GTA5 to Cityscapes based on ResNet-101.

Method	Adapt	Oracle	mIoU Gap
without adaptation	36.6	65.1	-28.5
global	39.3	65.1	-25.8
"soft" class-wise	38.0	65.1	-27.1
"soft" class-wise + global	40.2	65.1	-24.9
SS-D†	42.8	65.1	-22.3
SS-D† + global	42.6	65.1	-22.5
SS-D	44.0	65.1	-21.1
SS-D + CA-R	45.4	65.1	-19.7

5. Conclusion

In this paper, we tackle the domain adaptation problem for semantic segmentation via independent class-wise adversarial learning. We investigate an SS-D to address the critical issue of inconsistent adaptation and present a CA-R module to balance the class-wise adversarial learning process. Experimental results demonstrate that our approach outperforms the state-of-the-arts by considerable margins. With regard to some scenes with dense and small objects, our model has limited capability to distinguish them. In the future work, we plan to investigate this problem and extend our approach to more applications.

Acknowledgement

Project supported by the Shanghai Municipal Science and Technology Major Project (Grant No. 2018SHZDZX01, ZHANGJIANG LAB) and National Natural Science Foundation of China (Grant No. 61806189).

References

[1] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

[2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018.

[4] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[5] Yuhua Chen, Wen Li, and Luc Van Gool. Road: Reality oriented adaptation for semantic segmentation of urban scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7892–7901, 2018.

[6] Yi-Hsin Chen, Wei-Yu Chen, Yu-Ting Chen, Bo-Cheng Tsai, Yu-Chiang Frank Wang, and Min Sun. No more discrimination: Cross city adaptation of road scene segmenters. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1992–2001, 2017.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. 2009.

[8] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*, 2014.

[9] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[10] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012.

[11] Ankur Handa, Viorica Patraucean, Vijay Badrinarayanan, Simon Stent, and Roberto Cipolla. Understanding real world indoor scenes with synthetic data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4077–4085, 2016.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[14] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. *arXiv preprint arXiv:1711.03213*, 2017.

[15] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv preprint arXiv:1612.02649*, 2016.

[16] Weixiang Hong, Zhenzhen Wang, Ming Yang, and Junsong Yuan. Conditional generative adversarial network for structured domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1335–1344, 2018.

[17] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[18] Trung Le, Khanh Nguyen, and Dinh Phung. Theoretical perspective of deep domain adaptation. *arXiv preprint arXiv:1811.06199*, 2018.

[19] Peilun Li, Xiaodan Liang, Daoyuan Jia, and Eric P Xing. Semantic-aware grad-gan for virtual-to-real urban scene adaption. *arXiv preprint arXiv:1801.01726*, 2018.

[20] Guosheng Lin, Chunhua Shen, Anton Van Den Hengel, and Ian Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3194–3203, 2016.

[21] Ziwei Liu, Xiaoxiao Li, Ping Luo, Chen-Change Loy, and Xiaoou Tang. Semantic image segmentation via deep parsing network. In *Proceedings of the IEEE international conference on computer vision*, pages 1377–1385, 2015.

[22] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[23] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*, 2015.

[24] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems*, pages 136–144, 2016.

[25] Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2507–2516, 2019.

[26] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. 2009.

[27] Batch Normalization. Accelerating deep network training by reducing internal covariate shift. *CoRR.–2015.–Vol. abs/1502.03167.–URL: http://arxiv. org/abs/1502.03167*, 2015.

[28] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *European Conference on Computer Vision*, pages 102–118. Springer, 2016.

[29] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[30] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3234–3243, 2016.

[31] Swami Sankaranarayanan, Yogesh Balaji, Arpit Jain, Ser Nam Lim, and Rama Chellappa. Learning from synthetic data: Addressing domain shift for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3752–3761, 2018.

[32] Kihyuk Sohn, Sifei Liu, Guangyu Zhong, Xiang Yu, Ming-Hsuan Yang, and Manmohan Chandraker. Unsupervised domain adaptation for face recognition in unlabeled videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3210–3218, 2017.

[33] Kevin Tang, Vignesh Ramanathan, Li Fei-Fei, and Daphne Koller. Shifting weights: Adapting object detectors from image to video. In *Advances in Neural Information Processing Systems*, pages 638–646, 2012.

[34] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7472–7481, 2018.

[35] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4068–4076, 2015.

[36] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7167–7176, 2017.

[37] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2517–2526, 2019.

[38] Zuxuan Wu, Xintong Han, Yen-Liang Lin, Mustafa Gokhan Uzunbas, Tom Goldstein, Ser Nam Lim, and Larry S Davis. Dcan: Dual channel-wise alignment networks for unsupervised scene adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 518–534, 2018.

[39] Zifeng Wu, Chunhua Shen, and Anton Van Den Hengel. Wider or deeper: Revisiting the resnet model for visual recognition. *Pattern Recognition*, 90:119–133, 2019.

[40] Guorun Yang, Hengshuang Zhao, Jianping Shi, Zhidong Deng, and Jiaya Jia. Segstereo: Exploiting semantic information for disparity estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 636–651, 2018.

[41] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[42] Jing Zhang, Wanqing Li, and Philip Ogunbona. Transfer learning for cross-dataset recognition: a survey. *arXiv preprint arXiv:1705.04396*, 2017.

[43] Yang Zhang, Philip David, and Boqing Gong. Curriculum domain adaptation for semantic segmentation of urban scenes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2020–2030, 2017.

[44] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.

[45] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1529–1537, 2015.

[46] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2223–2232, 2017.

[47] Xinge Zhu, Hui Zhou, Ceyuan Yang, Jianping Shi, and Dahua Lin. Penalizing top performers: Conservative loss for semantic segmentation adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 568–583, 2018.

[48] Xiaojin Jerry Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2005.

[49] Yang Zou, Zhiding Yu, BVK Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 289–305, 2018.