Style Mixing and Patchwise Prototypical Matching for One-Shot Unsupervised Domain Adaptive Semantic Segmentation

Xinyi Wu[1], **Zhenyao Wu**[1], **Yuhang Lu**[1], **Lili Ju**[2,*], **Song Wang**[1,*]

[1]Department of Computer Science and Engineering, University of South Carolina, USA
[2]Department of Mathematics, University of South Carolina, USA
{xinyiw, zhenyao, yuhang}@email.sc.edu, ju@math.sc.edu, songwang@cec.sc.edu

Abstract

In this paper, we tackle the problem of one-shot unsupervised domain adaptation (OSUDA) for semantic segmentation where the segmentors only see one unlabeled target image during training. In this case, traditional unsupervised domain adaptation models usually fail since they cannot adapt to the target domain with over-fitting to one (or few) target samples. To address this problem, existing OSUDA methods usually integrate a style-transfer module to perform domain randomization based on the unlabeled target sample, with which multiple domains around the target sample can be explored during training. However, such a style-transfer module relies on an additional set of images as style reference for pre-training and also increases the memory demand for domain adaptation. Here we propose a new OSUDA method that can effectively relieve such computational burden. Specifically, we integrate several style-mixing layers into the segmentor which play the role of style-transfer module to stylize the source images without introducing any learned parameters. Moreover, we propose a patchwise prototypical matching (PPM) method to weighted consider the importance of source pixels during the supervised training to relieve the negative adaptation. Experimental results show that our method achieves new state-of-the-art performance on two commonly used benchmarks for domain adaptive semantic segmentation under the one-shot setting and is more efficient than all comparison approaches.

Introduction

Semantic segmentation is a basic computer vision task to identify the semantic category of each pixel from a set of pre-defined categories. It benefits a variety of tasks such as autonomous driving (Treml et al. 2016), medical imaging (Taghanaki et al. 2021), and image editing (Aksoy et al. 2018). Using deep learning, state-of-the-art semantic segmentation can be obtained in the form of good prediction at each pixel by training the well-designed segmentor network (Chen et al. 2017a) on large-scale labeled datasets and testing on the same domain. However, constructing datasets for such a dense prediction task is both very time-consuming and labor-intensive, which makes it often impossible to prepare a high-quality large-scale labeled training set for all different scenarios/domains, e.g., different cities or different illumination conditions. As a result, the generalization ability of a trained model is limited, i.e., it usually suffers from a drastic performance drop on an unseen testing domain due to the different data distributions from the training set.

Recently, some unsupervised domain adaptation approaches were proposed to overcome the domain discrepancy and reduce the demand for labeled data in new unseen test domains. Synthetic-to-real is a common setting in domain adaptive semantic segmentation, which was first proposed by Hoffman et al. (2016). In this setting, source domains with labeled synthetic data (Richter et al. 2016; Ros et al. 2016) are constructed by using computer graphics techniques and in the meantime, a sufficient number of samples in the target domain are also provided without labels. In many applications, such as medical imaging, a large collection of unlabeled target data may be unavailable or difficult to obtain, which leads to the introduction of a new setting, the one-shot unsupervised domain adaptation (OSUDA) (Luo et al. 2020) for semantic segmentation. The difference of the general unsupervised domain adaptation (UDA), domain generalization (DG) and OSUDA is illustrated in Fig. 1. In this paper, we aim to tackle this challenging but practical setting of OSUDA.

Existing UDA approaches, especially those which employ discriminators to distinguish whether the content, e.g., image feature (Hoffman et al. 2016), segmentation prediction (Tsai et al. 2018) or entropy map (Vu et al. 2019), is from the source or target domains (Fig. 1(a)), are prone to over-fitting on only one target sample – discriminators can easily distinguish the over-fit target domain from the source domain. Other style-transfer-based approaches cannot handle this one-shot setting either, since the source images can only be stylized by only one target sample. To solve this problem, Luo et.al proposed an adversarial style mining (ASM) algorithm (Luo et al. 2020), as illustrated in Fig. 1(c), by mutually optimizing the style-transfer module and the semantic segmentation network via an adversarial regime. However, the style-transfer module itself requires additional data for pre-training and also increases the demand for GPU memory for adaptation.

In this paper, we propose a new OSUDA approach, as illustrated in Fig. 1(d), which does not require additional data to pre-train a style-transfer module and explicitly synthesizes stylized images for semantic segmentation. First, we design

*Co-corresponding authors. Code is available at https://github.com/W-zx-Y/SM-PPM.

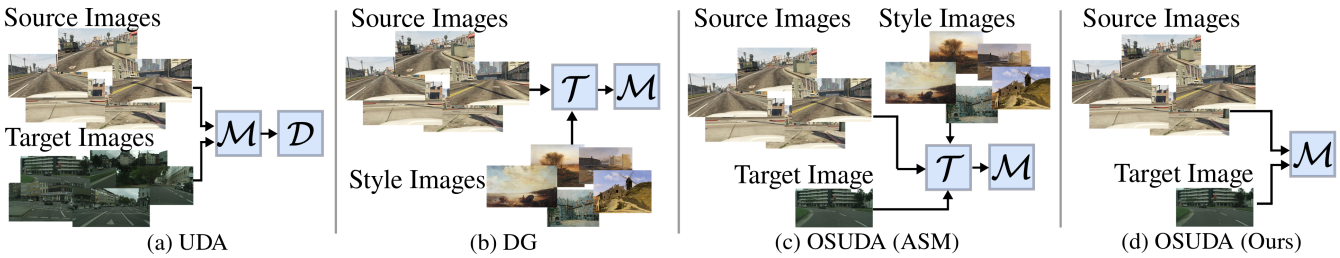

Figure 1: An illustration of general UDA, DG and OSUDA for semantic segmentation. The difference is mainly in the number of unlabeled target samples that are used for adaptation. Here, $\mathcal{M}$, $\mathcal{D}$ and $\mathcal{T}$ represent the segmentor, discriminator and style-transfer module, respectively.

a style-mixing segmentor which can simultaneously augment the source domain conditioned on feature statistics of the target sample and produce the semantic segmentation results. In addition, to relieve the negative adaptation (Li et al. 2020), i.e., not all source samples/pixels have a positive effect for domain adaptation, the source images are weightedly trained based on their similarity with patchwise prototypes of the sole target sample during domain adaptation.

The main contributions of this paper are summarized as follows. We propose a simple and effective method for OSUDA semantic segmentation, which makes full use of the sole target image in two aspects: (1) implicitly stylizing the source domain in both image and feature levels; (2) softly selecting the source training pixels. No additional images and training parameters are introduced in the whole process. It is worth mentioning that, with a pre-trained model on the source domain, our method only needs 20 minutes (500 iterations) to adapt to the target domain and obtains comparable results to the current best OSUDA approach (200k iterations without a pre-trained model, and additional training iterations for style-transfer model). Experimental results on two commonly-used synthetic-to-real scenarios demonstrate the effectiveness and efficiency of the proposed method.

Related Work

In this section, we briefly review the previous related works on UDA/OSUDA, DG, prototypical representation and style transfer, especially their applications to semantic segmentation.

Unsupervised Domain Adaptation and Domain Generalization

OSUDA is developed from the general UDA setting. The first UDA approach for semantic segmentation was proposed in Hoffman et al. (2016) using feature-level adversarial learning and category-specific adaptation. After that, adversarial learning has been applied to UDA in feature level (Chen et al. 2017b; Hoffman et al. 2018), output space (Tsai et al. 2018; Pan et al. 2020; Zhang, David, and Gong 2017; Perone et al. 2019) and entropy of the prediction (Vu et al. 2019; Pan et al. 2020) for alignment. Image translation is another approach for UDA (Sankaranarayanan et al. 2018; Li, Yuan, and Vasconcelos 2019; Yang and Soatto 2020) by exploiting advanced image-to-image translation networks, e.g., CycleGAN (Zhu et al. 2017), to reduce the domain discrepancy. Recently, multiple rounds of self-training with generated pseudo labels of the target domain samples was proved to be a powerful strategy to boost the adaptation performance (Zou et al. 2018; Li, Yuan, and Vasconcelos 2019; Zhang et al. 2019). However, these methods cannot be directly applied to the OSUDA setting due to the scarce of the target images.

Also related to OSUDA is the problem of domain generalization where the target domain is totally unknown. Based on the number of source domains involved during adaptive learning, existing DG approaches can be basically divided into multi-source DG (Gong, Grauman, and Sha 2013; Dou et al. 2019) and single-source DG (Pan et al. 2018; Yue et al. 2019; Huang et al. 2021). For multi-source DG, Zhou et.al proposed a MixStyle (Zhou et al. 2021) strategy to increase the domain diversity of the source domains. During training, two instances of different domains in a mini-batch are selected to synthesize novel domains leveraging the feature-level style statistics (Huang and Belongie 2017a). Single-source DG is more challenging since less labeled source data is accessible for adaptation. A typical solution is to perform domain randomization (Tobin et al. 2017) on the source training samples via image stylization or translation which can also be treated as a data/domain augmentation strategy. For example, several real-life images from ImageNet (Deng et al. 2009) are picked as randomization references (Yue et al. 2019) to adjust the source images or their domain invariant frequency components (Huang et al. 2021).

Prototypical Representation

Prototypes are defined as abstractions of essential semantic feature representations, which were popularly used in computer vision tasks recently. For example, Wang et al. (2019) design a prototype alignment regularization for few-shot semantic segmentation, where the class-specific prototypes are computed via a masked average pooling. More recently, Zhang et al. (2021) exploited the distances between the target features and the class-wise prototypes to re-weight the predicted probability for better self-training. In this paper, we calculate the prototypes of the patches of the sole target image to weigh the training pixels from the source domain.

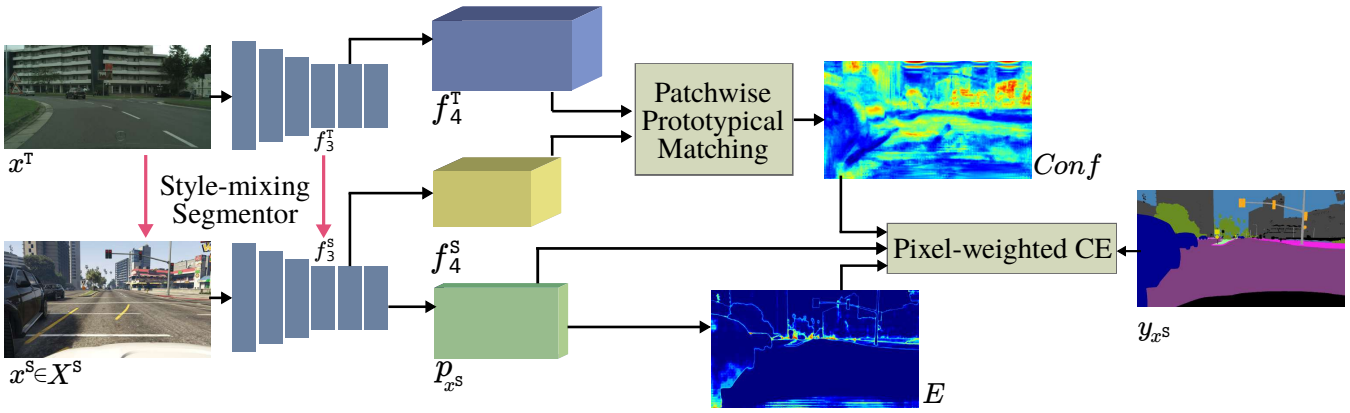

Figure 2: An illustration of the proposed method for OSUDA semantic segmentation. The pink arrows indicate the positions that style-mixing operation is performed.

Style Transfer

Gatys, Ecker, and Bethge (2015) proposed a neural style-transfer algorithm to generate high-quality artistic images by separating and recombining the content and style of arbitrary images. Later style transfer has become an effective technique, which benefits several real-world applications such as makeup transfer and removal (Chang et al. 2018) and virtual try-on (Yang et al. 2020). Our work is closely related to the adaptive instance normalization (AdaIN) proposed by Huang and Belongie (2017b), which transfers the mean and variance in the feature space in real-time. The main difference is that we don't synthesize the image with a decoder.

Our Approach

Given labeled samples $(X^{\mathrm{S}}, Y^{\mathrm{S}})$ from the source domain S and unlabeled samples X^{T} from the target domain T, the goal of general UDA problem is to learn a mapping $\mathcal{G}$ formulated as

$$\mathcal{G}(X^{\mathrm{S}}) \rightarrow Y^{\mathrm{S}}; \quad \mathcal{F}(\mathrm{S}) \rightarrow \mathcal{F}(\mathrm{T}), \tag{1}$$

where $\mathcal{F}$ represents any function to align the two domains, e.g., a discriminator. Different from general UDA, only one unlabeled target sample $x^{\mathrm{T}} \in X^{\mathrm{T}}$ is accessible in the OSUDA setting, which can be formulated as:

$$\mathcal{G}(X^{\mathrm{S}}|x^{\mathrm{T}}) \rightarrow Y^{\mathrm{S}}. \tag{2}$$

The network architecture of the proposed one-shot unsupervised domain adaptive semantic segmentation method is illustrated in Fig. 2, which is composed of a style-mixing segmentor for both style transfer and semantic segmentation, and a patchwise prototypical matching module for weighting the pixels of the source domain. The details of the two main components and the objective functions are discussed below.

Style-mixing Segmentor

For each iteration, the style-mixing segmentor first takes the target sample x^{T} as input in the evaluation mode to achieve target features using the current model parameters. Then, a sample x^{S} is randomly chosen from the source domain and fed into the segmentor in the training mode.

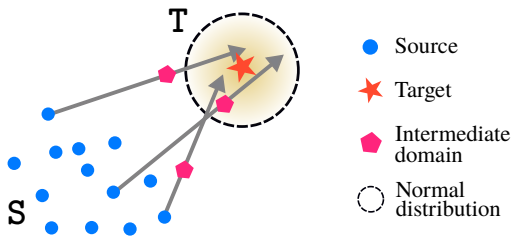

Figure 3: An illustration of the style-mixing operation. We first augment the target feature statistics by adding a perturbation sampled from normal distribution. Then some intermediate domains can be obtained by mixing the feature statistics of the source and the augmented target.

Inspired by Zhou et al. (2021), we mix the target features from the sole target image with the source features via instance normalization layers to obtain intermediate domain features.

Following Huang and Belongie (2017b), we compute the channel-wise spatial mean $\mu(.)$ and standard deviation $\sigma(.)$ of any given sample/feature $f \in \mathbb{R}^{C \times H \times W}$ via

$$\mu(f) = \frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} f \tag{3}$$

and

$$\sigma(f) = \sqrt{\frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} (f - \mu(f))^2 + \epsilon}, \tag{4}$$

where C, H and W are channel, height and width of f, respectively. ϵ is set to 10^{-30}. Since the only one target sample is insufficient to describe the whole target feature distribution, we exploit more feature statistics centered around f^{T} as shown in Fig. 3. Then, we mix the statistics of source and target domains and calculate the intermediate channel-wise mean γ and standard deviation β by

$$\gamma = \lambda \sigma(f^{\mathrm{S}}) + (1 - \lambda) \left(\sigma(f^{\mathrm{T}}) + r_{\sigma}\right), \tag{5}$$

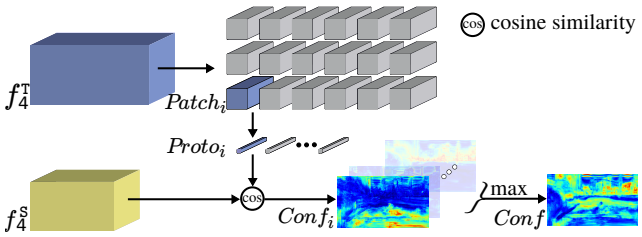

Figure 4: An illustration of the proposed patchwise prototypical matching.

$$\beta = \lambda\mu(f^{\mathrm{S}}) + (1-\lambda)\left(\mu(f^{\mathrm{T}}) + r_\mu\right), \tag{6}$$

where $\lambda \in \mathbb{R}^C$ are weights to balance the mixing operation which are randomly sampled from uniform distribution for each image/feature pair, $f^{\mathrm{S}} \in \{x^{\mathrm{S}}, f_3^{\mathrm{S}}\}$ and $f^{\mathrm{T}} \in \{x^{\mathrm{T}}, f_3^{\mathrm{T}}\}$ with f_3^{S} and f_3^{T} denoting the source and target features achieved from *layer3* respectively. Here we take $r_\sigma \sim N(0, \frac{|\sigma(f^{\mathrm{T}}) - \sigma(f^{\mathrm{S}})|}{10})$ and $r_\mu \sim N(0, \frac{|\mu(f^{\mathrm{T}}) - \mu(f^{\mathrm{S}})|}{10})$. The stylized source feature $\widehat{f^{\mathrm{S}}}$ is then produced by taking

$$\widehat{f^{\mathrm{S}}} = \gamma\left(\frac{f^{\mathrm{S}} - \mu(f^{\mathrm{S}})}{\sigma(f^{\mathrm{S}})}\right) + \beta. \tag{7}$$

Patchwise Prototypical Matching

Li et al. (2020) empirically found that some source samples could have negative effect on the adaptation. Based on this observation, they perform both image and pixel-level selections in the source domain to avoid the negative domain adaptation. However, both of their image and pixel-level selections are dependent on the distribution analysis of the target domain predictions, which are not applicable to our one-shot setting, i.e., the sole target image cannot correctly reflect the data distribution in the target domain and many categories are missing in this target sample. Inspired by Li et al. (2020), we propose a patchwise prototypical matching (PPM) by softly adjusting the weight of each source pixel during training according to their similarity with the target sample. We do not perform image-level selection since "negative" samples might also contain "positive" pixels for adaptation.

Specifically, we reshape the target image feature $f_4^{\mathrm{T}} \in \mathbb{R}^{C_4 \times H_4 \times W_4}$ obtained from *layer4* of the segmentor into the form of patches $p_4^{\mathrm{T}} \in \mathbb{R}^{N \times C_4 \times P^2}$ as shown in Fig. 4, where H_4, W_4, C_4 are the height, width and number of channels of f_4^{T}, P is the patch size and N is the number of patches. There is no overlap between two patches. Then, we compute the prototype for each patch via:

$$Proto_i = \frac{1}{P^2}\sum_{s=1}^{P}\sum_{t=1}^{P} Patch_i(s,t), \tag{8}$$

where $Proto_i \in \mathbb{R}^{C_4}$, $i \in [0, N-1]$ and (s,t) specifies each position in the patch. We compute the similarity between each prototype and the source features as a confidence map $Conf_i \in \mathbb{R}^{C_4 \times H_4 \times W_4}$ for adaptation by

$$Conf_i = \mathcal{F}\left(f_4^{\mathrm{S}}, Proto_i\right), \tag{9}$$

Algorithm 1: Patchwise Prototypical Matching

Input: Source images X^{S}; source labels Y^{S}; one-shot target image x^{T}; style-mixing segmentor $\mathcal{M}$ with the parameter θ and learning rate lr

Output: Optimal θ^*
for $x^{\mathrm{S}} \in X^{\mathrm{S}}$ **do**
  With no gradients:
    $(p_{x^{\mathrm{T}}}, f_3^{\mathrm{T}}, f_4^{\mathrm{T}}) = \mathcal{M}\left(x^{\mathrm{T}}, \text{style} = \text{None}\right)$;
    $Patch$ = rearrange(f_4^{T});
    Compute $Proto$ via Eq.(8);
  $(p_{x^{\mathrm{S}}}, f_3^{\mathrm{S}}, f_4^{\mathrm{S}}) = \mathcal{M}\left(x^{\mathrm{S}}, \text{style} = (x^{\mathrm{T}}, f_3^{\mathrm{T}})\right)$;
  Compute the E using the $p_{x^{\mathrm{S}}}$ via Eq. (11);
  Compute the $\widehat{Conf}$ using f_4^{S} and $Patch$ via Eqs. (9), (10) and (12);
  Update the parameter:
    $\theta \leftarrow \theta - lr\nabla_\theta \mathcal{L}(p_{x^{\mathrm{S}}}, y_{x^{\mathrm{S}}}, \widehat{Conf}, E)$;
end for
Return θ as θ^*

where $f_4^{\mathrm{S}} \in \mathbb{R}^{C_4 \times H_4 \times W_4}$ is the source image feature obtained from *layer4*, and we choose the cosine similarity as the distance function $\mathcal{F}$. We then perform a max operation across all prototypes to obtain the $Conf \in R^{H_4 \times W_4}$ for this source sample in this running iteration by

$$Conf = \max_{i \in [0, N-1]} Conf_i. \tag{10}$$

The reason for using the prototypical representation of the target features to compute the confidence maps include: 1) it is more efficient than pixel-wise similarity computation; 2) due to the domain gap, the pixel-level similarity usually contains much more noise which can be relieved by using patchwise prototypes. Finally, the confidence map is rectified based on the entropy of the source prediction. Given the source prediction $p_{x^{\mathrm{S}}}$, its entropy map $E \in \mathbb{R}^{H_4 \times W_4}$ can be achieved via:

$$E = -\frac{1}{\log(C)}\sum_{c=1}^{C}\left(p_{x^{\mathrm{S}}}^{(c)} \cdot \log(p_{x^{\mathrm{S}}}^{(c)})\right), \tag{11}$$

where C is the number of classes. Through this way, the rectified confidence map is achieved by

$$\widehat{Conf} = Conf \cdot (1 - E). \tag{12}$$

High entropy indicates low confidence for the prediction, therefore, $(1-E)$ can highlight the confident region based on the prediction. The thought behind this design is that the source should be confident enough to help the adaptation to the target. A detailed pipeline for the proposed PPM is given in Algorithm 1.

Objective Functions

In general, the semantic segmentation task applies the cross entropy as the loss function:

$$\mathcal{L}_{ce} = -\frac{1}{HW}\sum_{h,w}^{HW}\sum_{c=1}^{C}\left(y_{x^{\mathrm{S}}}^{(h,w,c)} \cdot \log(p_{x^{\mathrm{S}}}^{(h,w,c)})\right), \tag{13}$$

Method	Extra	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorcycle	bicycle	**mIoU**
Source only	-	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
Adaptseg	O	77.7	19.2	75.5	11.7	6.4	16.8	18.2	15.4	77.1	34.0	68.5	55.3	30.9	74.5	23.7	28.3	2.9	14.4	18.9	35.2
CLAN	O	77.1	22.7	78.6	17.0	14.8	20.5	23.8	12.0	80.2	39.5	74.3	56.6	25.2	78.1	29.3	31.2	0.0	19.4	16.7	37.7
ADVENT	O	76.1	15.1	76.6	14.4	10.8	17.5	19.8	12.0	79.2	39.5	71.3	55.7	25.2	76.7	28.3	30.5	0.0	23.6	14.4	36.1
CBST	O	76.1	22.2	73.5	13.8	18.8	19.1	20.7	18.6	79.5	41.3	74.8	57.4	19.9	78.7	21.3	28.5	0.0	28.0	13.2	37.1
CycleGAN	O	80.3	23.8	76.7	17.3	18.2	18.1	21.3	17.5	81.5	40.1	74.0	56.2	**38.3**	77.1	30.3	27.6	1.7	**30.0**	22.2	39.6
OST	O	84.3	**27.6**	**80.9**	**24.1**	23.4	26.7	23.2	19.4	80.2	**42.0**	**80.7**	**59.2**	20.3	**84.1**	**35.1**	**39.6**	1.0	29.1	23.2	42.3
Ours	O	**85.0**	23.2	80.4	21.3	**24.5**	**30.0**	**32.0**	**26.7**	**83.2**	34.8	74.0	57.3	29.0	77.7	27.3	36.5	**5.0**	28.2	**39.4**	**42.8**
DRPC	S	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	42.5
FSDR	S	89.3	40.5	79.1	26.3	27.8	29.3	33.7	29.0	83.0	27.7	76.0	57.8	27.5	81.0	32.3	42.4	16.8	21.0	30.2	44.8
ASM	O+S	89.5	31.2	81.3	27.8	22.8	30.6	32.8	25.1	82.6	35.0	76.7	59.2	26.6	82.3	27.7	34.1	0.9	25.6	29.6	43.2

Table 1: Quantitative comparison results for domain adaptation from GTA5 to Cityscapes. The per-category mIoU (%) of the Cityscapes-val set are reported. For all methods with one-shot only setting denoted by O, the best results are presented in bold, with the second best results underlined.

where $y_{x^s}^{(h,w,c)}$ represents the one-hot encoding of the ground-truth label at position (h, w) for the class c. In our approach, we employ the final confidence map $\widehat{Conf}$ to adjust the weight of each source sample in pixel-level via

$$\mathcal{L}_{pce} = -\frac{1}{HW}\sum_{h,w}^{HW}(\widehat{Conf}^{(h,w)} \cdot \sum_{c=1}^{C}(\, y_{x^s}^{(h,w,c)} \cdot \log(p_{x^s}^{(h,w,c)}))). \tag{14}$$

Finally, the whole network is trained with

$$\mathcal{L} = \alpha\mathcal{L}_{ce} + \mathcal{L}_{pce}, \tag{15}$$

where α is the balancing factor which is set to 0.5 in all experiments.

Experiments

Datasets and Evaluation Metric

We evaluate the proposed OSUDA semantic segmentation method in two synthetic-to-real scenarios, i.e., GTA5 (Richter et al. 2016) $\rightarrow$ Cityscapes (Cordts et al. 2016) and SYNTHIA (Ros et al. 2016) $\rightarrow$ Cityscapes. Both GTA5 and SYNTHIA datasets are treated as the source domains, where the former contains 24,966 images with a resolution of $1,914 \times 1,052$ and the latter contains 9,400 images with a resolution of $1,280 \times 760$. We use Cityscapes as the target domain which is split into 2,975/500/1,525 images for training/validation/testing purposes. We follow the one-shot setting in Luo et al. (2020) where only one unlabeled target image is used for domain adaptation. In GTA5 $\rightarrow$ Cityscapes, 19 common categories are evaluated and in SYNTHIA $\rightarrow$ Cityscapes, 16 common categories are evaluated. We apply the Intersection over Union (IoU) as the evaluation metric.

Implementation Details

The proposed method is implemented using PyTorch trained on a single Nvidia 2080Ti GPU. We use the DeepLabV2-Res101 (Chen et al. 2017a) initialized with the source-only trained weights provided by Tsai et al. (2018) as the segmentor. The source images are resized to $1,280 \times 760$ and the one-shot target sample keeps its original size. We train the network using the SGD (Bottou 2010) optimizer with a momentum of 0.9 and a weight decay of 5×10^{-4}. The initial learning rate is set to 2.5×10^{-5} and it is decreased gradually following the poly learning rate policy in Tsai et al. (2018). The batch size is set to 1 and the whole network is trained for 500 iterations. Note that we even don't get access to all source images during domain adaptation. Images from Cityscapes validation set are resized to $1,024 \times 512$ for performance evaluation. We run each OSUDA experiment with the same 5 images as Luo et al. (2020) (one for each time) and 5 times for each image. Finally, we report the average mIoU of the 25 runs computed using the model weights saved in the last running iteration. All the approaches are evaluated on the Cityscapes-val set.

Comparison Results

In Table 1, we present the comparison results with other state-of-the-art approaches for the GTA5 $\rightarrow$ Cityscapes experiment. The compared method can be divided into three camps based on the data samples except for the source domain that are needed for adaptation: 1) one unlabeled target image only (denoted by O); 2) style image dataset (denoted by S); (3) both 1) and 2) (denoted by O+S). It can be observed that our method achieves the best performance in the first camp. Obviously, the general UDA approaches Adaptseg (Tsai et al. 2018), CLAN (Luo et al. 2019) and CBST (Zou et al. 2018) are not working well in the one-shot setting and some even get worse results than the source only. Methods CycleGAN (Zhu et al. 2017) and OST (Benaim and Wolf 2018) are proved to

Method	Extra	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	sky	person	rider	car	bus	motorcycle	bicycle	**mIoU**	**mIoU***
Source only	-	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	-
Adaptseg	O	64.1	25.6	75.3	-	-	-	4.7	2.7	77.0	70.0	52.2	20.6	51.3	22.4	19.9	22.3	39.1	-
CLAN	O	68.3	26.9	72.2	-	-	-	5.1	5.3	75.9	71.4	54.8	18.4	65.3	19.2	22.1	20.7	40.4	-
ADVENT	O	65.7	22.3	69.2	-	-	-	2.9	3.3	76.9	69.2	**55.4**	21.4	**77.3**	17.4	21.4	16.7	39.9	-
CBST	O	59.6	24.1	72.9	-	-	-	5.5	13.8	72.2	69.8	55.3	21.1	57.1	17.4	13.8	18.5	38.5	-
OST	O	75.3	31.6	72.1	-	-	-	12.3	9.3	76.1	71.1	51.1	17.7	68.9	19.0	**26.3**	25.4	42.8	-
Ours	O	**79.3**	**35.3**	**75.9**	5.6	16.6	29.8	**25.4**	**22.7**	**79.9**	**76.8**	54.6	**23.5**	60.2	**23.9**	21.2	**36.6**	**47.3**	**41.4**
DRPC	S	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	37.6
FSDR	S	69.3	34.9	77.6	7.9	0.2	29.4	16.3	19.2	72.3	76.3	56.7	22.1	80.6	41.5	19.1	29.3	47.3	40.8
ASM	O+S	85.7	39.7	77.1	1.1	0.0	24.2	2.1	9.2	76.9	81.7	43.4	11.4	63.9	15.8	1.6	20.3	40.7	34.6

Table 2: Quantitative comparison results for domain adaptation from SYNTHIA to Cityscapes. The per-category mIoU (%) (13 categories) and mIoU* (%) (16 categories) of Cityscapes-val set are reported. For all methods with one-shot only setting denoted by O, the best results are presented in bold, with the second best results underlined.

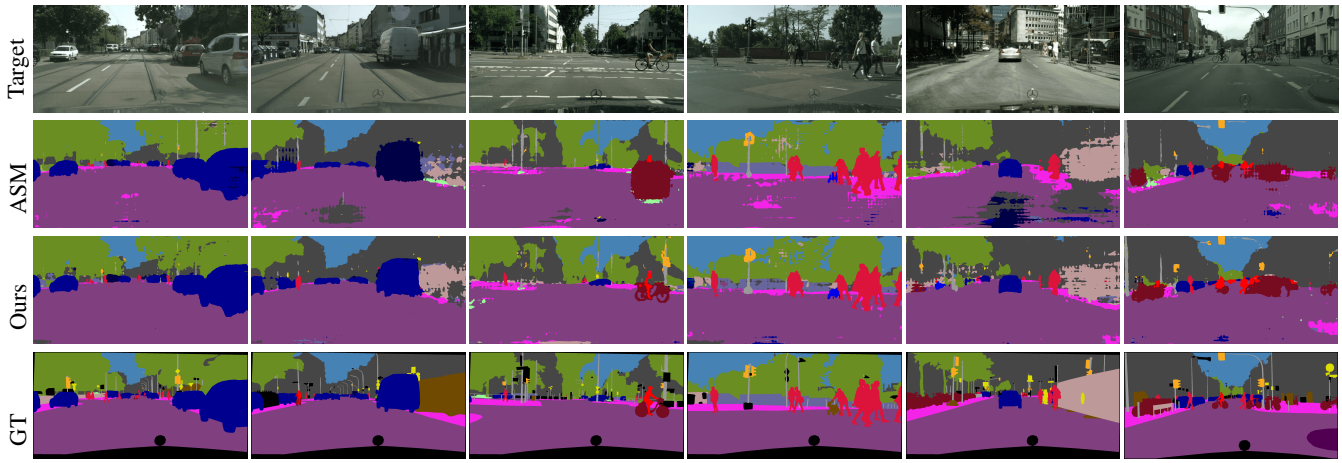

Figure 5: Some qualitative comparison results for domain adaptation from GTA5 → Cityscapes.

be more robust to this setting which indicates the usefulness of the style transfer strategy. ASM (Luo et al. 2020) is the first method that tackles the OSUDA which is the most related one to ours. To make a fair comparison, we reproduce the results of ASM using the same backbone[1] as us and the reported mIoU is also based on the model saved in the last iteration (not selecting the best one). Especially, our method does not need an additional dataset to pre-train a style transfer model while ASM needs, and ours runs only for 500 iterations for domain adaptation to achieve these comparable results. We also find that domain generalization approaches DRPC (Yue et al. 2019) and FSDR (Huang et al. 2021) using additional style image datasets also achieve comparable results or even better than the methods using one target image. This indicates that using more images than only one target image can be more helpful as expected. However, they need to spend more time exploring the desired domains and the style references also need to be properly chosen. The results for SYNTHIA → Cityscapes experiment are reported in Table. 2, where our method achieves the best performance across all of the three settings and surpasses the second-best by 4.5% mIoU in the one-shot only setting.

We also show qualitative results for GTA5 → Cityscapes and SYNTHIA → Cityscapes each on 5 samples from the Cityscapes-val set in Fig. 5 and Fig. 6, respectively. It can be observed that our method achieves comparable visualization results as ASM in the two domain adaptation scenarios and even better on some categories such as train (Fig. 6(a)), rider and bicycle (Fig. 5(c)) and truck (Fig. 5(d)).

[1]https://github.com/RoyalVane/ASM/issues/2

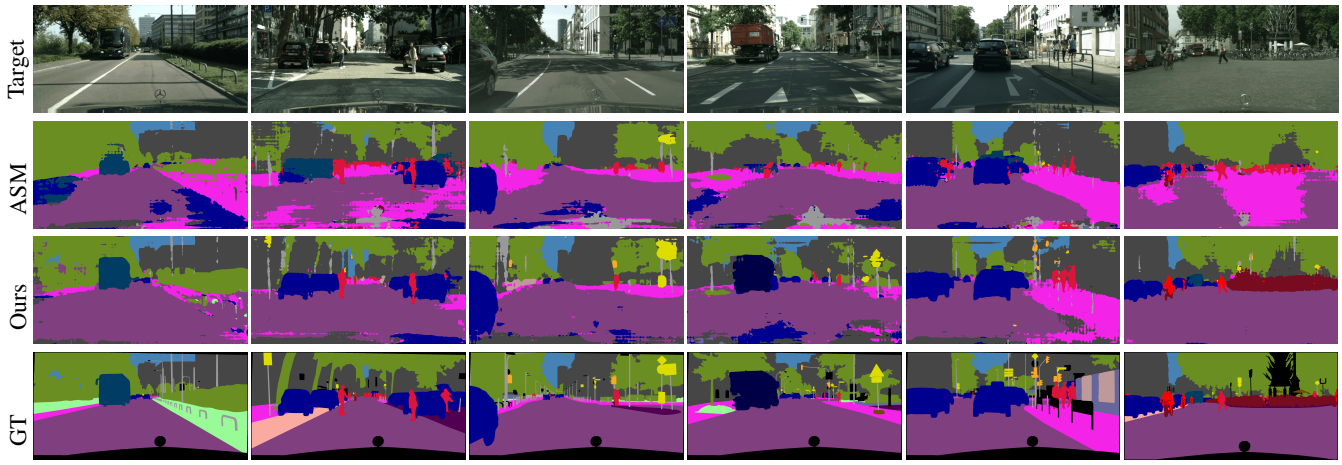

Figure 6: Some qualitative comparison results for domain adaptation from SYNTHIA → Cityscapes.

Method	Extra	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorcycle	bicycle	**mIoU**
Source only	-	69.5	12.4	44.9	3.0	18.5	17.8	15.6	7.0	34.2	7.4	8.4	12.3	0.8	22.8	0.0	0.0	0.0	3.4	0.2	14.6
ASM	O+S	75.2	31.7	38.6	7.7	17.6	16.9	12.6	4.9	24.4	8.0	9.8	16.7	1.4	42.9	0.0	0.0	0.0	7.7	2.1	16.7
Ours	O	63.3	16.6	46.5	5.1	22.9	9.0	15.5	7.7	39.7	10.1	31.5	10.2	0.7	38.7	0.0	0.0	0.0	11.0	4.2	17.5

Table 3: Quantitative comparison results for domain adaptation from Cityscapes to Dark Zurich. The per-category mIoU (%) of the Dark Zurich validation set are reported.

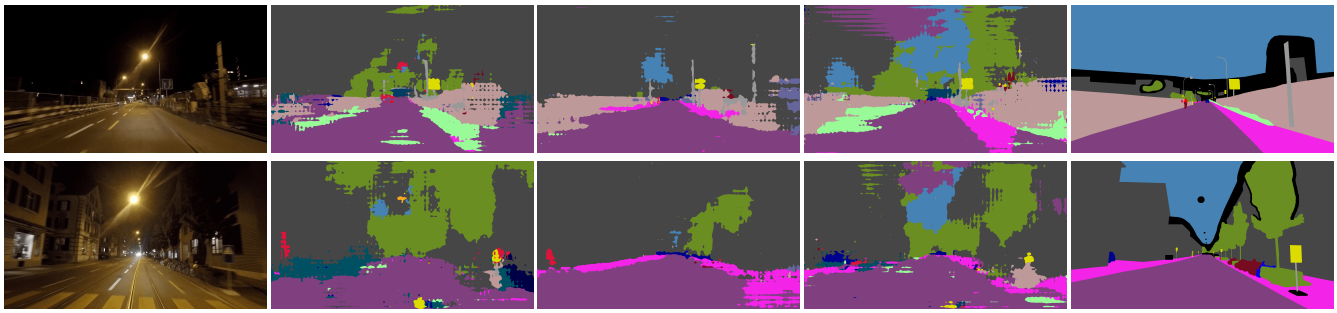

Figure 7: Some qualitative comparison results for domain adaptation from Cityscapes to Dark-Zurich.

One-Shot Day-to-Night Domain Adaptation

We further evaluate the proposed method on the more challenging day-to-night setting. In this experiment, we pick Cityscapes (Cordts et al. 2016) as the source and the Dark Zurich (Sakaridis, Dai, and Van Gool 2019) as the target.

The Dark Zurich dataset is carefully collected by Sakaridis et.al for unsupervised nighttime semantic segmentation. It consists of 3,041 daytime, 2,920 twilight and 2,416 nighttime images that are all unlabeled and can be used for domain adaptation. There are also 201 labeled nighttime images including 50 images for validation whose labels are provided and the rests serve as an online benchmark. The resolution of all images is 1,920 × 1,080.

In our experiments, only one nighttime image is used for domain adaptation and the Dark Zurich validation set is used for performance evaluation. We run this experiment with 4 images (one for each experiment) and 5 times for each image. Finally, we report the average mIoU of the 20 runs computed using the model weights saved in the last running iteration in Table 3. Here, the source-only model is obtained

by training the DeepLabV2-Res101 (Chen et al. 2017a) on the Cityscapes training set for 150K iterations. By applying the source-only model weights, our method can achieve 17.5% with an additional 500 training iterations. We run ASM with the same 4 images for 50K iterations and compute the average of the 4 experiments as their results. It can be observed that both of the two OSUDA approaches obtain performance gains over the source-only results and our method get better results without training an explicit style transfer model with additional dataset. Some qualitative results are shown in Fig. 7 where we can see that our method gets better visualization results.

Ablation Studies

Variants of the loss functions. We first investigate several variants of Eq. (12) as shown in Table. 4. We directly use the model weights provided by Tsai et al. (2018) and fine-tune it with the source data using the standard cross-entropy loss to obtain the source-only results. Note that all these variants are equipped with the original segmentor instead of the style-mixing one. We can observe that the confidence obtained via PPM is the most important component in this equation, without which the mIoU drops 2.66% on GTA5 $\rightarrow$ Cityscapes and 9.66% on SYNTHIA $\rightarrow$ Cityscapes. Compared with $\widehat{Conf}$, E has an inconsistent effect on the two experiments.

Variants	Source-only	w/o $\widehat{Conf}$	w/o E	Eq. (12)
G $\rightarrow$ C	36.67	38.38	40.95	**41.04**
S $\rightarrow$ C	35.26	34.26	**44.14**	43.92

Table 4: Variants of Eq. (12) in both GTA5 $\rightarrow$ Cityscapes (G $\rightarrow$ C) and SYNTHIA $\rightarrow$ Cityscapes (S $\rightarrow$ C) scenarios. The mIoU (%) scores are reported.

Variants of the style-mixing segmentor. We study several variants of the style-mixing segmentor as shown in Table. 5. The original Deeplab-V2-Res101 without style-mixing (with PPM) serves as the baseline which can obtain 41.04% mIoU. Applying the style-mixing layer in image-level (x^{S} only) can obtain 1.39% for GTA5 $\rightarrow$ Cityscapes and 3.04% for SYNTHIA $\rightarrow$ Cityscapes. In addition, we try different feature-level style-mixing and we observe that using f_3^{S} only is better than using other levels for both adaptation scenarios. Therefore, we choose to apply the style-mixing layer to both x^{S} and f_3^{S} (Ours). Other combinations might result in similar performance. Compared with the original version of AdaIN, our modified version achieves better performance.

Variants	base	x^{S}	f_1^{S}	f_2^{S}	f_3^{S}	f_4^{S}	AdaIN	Ours
G$\rightarrow$C	41.04	42.43	40.74	41.13	41.16	40.24	41.98	**42.77**
S$\rightarrow$C	43.92	46.96	43.65	43.71	44.63	43.92	46.82	**47.33**

Table 5: Variants of the style-mixing segmentor in both GTA5 $\rightarrow$ Cityscapes (G $\rightarrow$ C) and SYNTHIA $\rightarrow$ Cityscapes (S $\rightarrow$ C) scenarios. The mIoU (%) scores are reported.

Variants of the patch size. We also study different choices of patch size as shown in Table. 6, where "no patch" means that we don't split the target image into patches and use the prototype of the whole image to calculate the confidence map. We find that the patch size 32 performs the best size for both domain adaptation experiments.

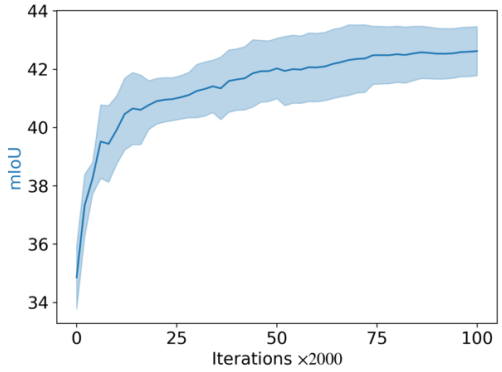

Figure 8: The mIoU (%) performance over varying adaptation iterations without using pertrained model for GTA5 $\rightarrow$ Cityscapes.

Patch size	8	16	32	64	no patch
G $\rightarrow$ C	42.43	42.30	**42.77**	42.38	42.26
S $\rightarrow$ C	44.83	45.91	**47.33**	45.52	46.03

Table 6: Variants of the patch size for both GTA5 $\rightarrow$ Cityscapes (G $\rightarrow$ C) and SYNTHIA $\rightarrow$ Cityscapes (S $\rightarrow$ C) scenarios. The mIoU (%) scores are reported.

Ablation Study on the pre-training model We study the effect of the usage of the pre-training model. From Fig. 8, we find that our method can still obtain similar mIoU results without using the pre-training model for GTA5 $\rightarrow$ Cityscapes with more training iterations. Compared with ASM, our method does not need additional dataset and time to train a style-transfer model and uses fewer adaptation iterations with a pre-trained source-only model. And our method can save the memory usage, for example, it only needs about 10G GPU memory while ASM requires around 25G.

Conclusion

In this paper, we have developed a novel method for the challenging one-shot semantic segmentation in unsupervised domain adaptation. The proposed style-mixing segmentor has the ability to explore more styles around the target sample and perform the semantic segmentation at the same time. This implicit style transfer based on feature-level statistics can significantly reduce memory usage and improve the efficiency of domain adaptation. In addition, patchwise prototypical matching, which is proposed for relieving the negative adaptation and weighting more the positive adaptation, is also shown to be very effective for this task. Various experiments demonstrate that our method can achieve better or comparable results to the current state-of-the-arts in the one-shot setting with much fewer iterations.

Acknowledgments

Dr. Lili Ju's work is partially supported by U.S. Department of Energy, Office of Advanced Scientific Computing Research through Applied Mathematics program under grant DE-SC0022254.

References

Aksoy, Y.; Oh, T.-H.; Paris, S.; Pollefeys, M.; and Matusik, W. 2018. Semantic soft segmentation. *ACM Transactions on Graphics (TOG)*, 37(4): 1–13.

Benaim, S.; and Wolf, L. 2018. One-Shot Unsupervised Cross Domain Translation. In *Advances in Neural Information Processing Systems*.

Bottou, L. 2010. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, 177–186. Springer.

Chang, H.; Lu, J.; Yu, F.; and Finkelstein, A. 2018. Pairedcyclegan: Asymmetric style transfer for applying and removing makeup. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 40–48.

Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L. 2017a. 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.

Chen, Y.-H.; Chen, W.-Y.; Chen, Y.-T.; Tsai, B.-C.; Frank Wang, Y.-C.; and Sun, M. 2017b. No more discrimination: Cross city adaptation of road scene segmenters. In *IEEE International Conference on Computer Vision (ICCV)*, 1992–2001.

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 248–255. IEEE.

Dou, Q.; Castro, D. C.; Kamnitsas, K.; and Glocker, B. 2019. Domain Generalization via Model-Agnostic Learning of Semantic Features. In *Advances in Neural Information Processing Systems*.

Gatys, L. A.; Ecker, A. S.; and Bethge, M. 2015. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*.

Gong, B.; Grauman, K.; and Sha, F. 2013. Reshaping visual datasets for domain adaptation. *Advances in Neural Information Processing Systems*, 26: 1286–1294.

Hoffman, J.; Tzeng, E.; Park, T.; Zhu, J.-Y.; Isola, P.; Saenko, K.; Efros, A.; and Darrell, T. 2018. Cycada: Cycle-consistent adversarial domain adaptation. In *International Conference on Machine Learning*, 1989–1998. PMLR.

Hoffman, J.; Wang, D.; Yu, F.; and Darrell, T. 2016. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv preprint arXiv:1612.02649*.

Huang, J.; Guan, D.; Xiao, A.; and Lu, S. 2021. FSDR: Frequency Space Domain Randomization for Domain Generalization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Huang, X.; and Belongie, S. 2017a. Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization. In *IEEE International Conference on Computer Vision (ICCV)*.

Huang, X.; and Belongie, S. 2017b. Arbitrary style transfer in real-time with adaptive instance normalization. In *IEEE International Conference on Computer Vision (ICCV)*, 1501–1510.

Li, G.; Kang, G.; Liu, W.; Wei, Y.; and Yang, Y. 2020. Content-consistent matching for domain adaptive semantic segmentation. In *European Conference on Computer Vision (ECCV)*, 440–456. Springer.

Li, Y.; Yuan, L.; and Vasconcelos, N. 2019. Bidirectional learning for domain adaptation of semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6936–6945.

Luo, Y.; Liu, P.; Guan, T.; Yu, J.; and Yang, Y. 2020. Adversarial Style Mining for One-Shot Unsupervised Domain Adaptation. In *Advances in Neural Information Processing Systems*.

Luo, Y.; Zheng, L.; Guan, T.; Yu, J.; and Yang, Y. 2019. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2507–2516.

Pan, F.; Shin, I.; Rameau, F.; Lee, S.; and Kweon, I. S. 2020. Unsupervised intra-domain adaptation for semantic segmentation through self-supervision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3764–3773.

Pan, X.; Luo, P.; Shi, J.; and Tang, X. 2018. Two at once: Enhancing learning and generalization capacities via ibn-net. In *European Conference on Computer Vision (ECCV)*, 464–479.

Perone, C. S.; Ballester, P.; Barros, R. C.; and Cohen-Adad, J. 2019. Unsupervised domain adaptation for medical imaging segmentation with self-ensembling. *NeuroImage*, 194: 1–11.

Richter, S. R.; Vineet, V.; Roth, S.; and Koltun, V. 2016. Playing for data: Ground truth from computer games. In *European Conference on Computer Vision (ECCV)*, 102–118. Springer.

Ros, G.; Sellart, L.; Materzynska, J.; Vazquez, D.; and Lopez, A. 2016. The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Sakaridis, C.; Dai, D.; and Van Gool, L. 2019. Guided Curriculum Model Adaptation and Uncertainty-Aware Evaluation for Semantic Nighttime Image Segmentation. In *IEEE International Conference on Computer Vision (ICCV)*.

Sankaranarayanan, S.; Balaji, Y.; Jain, A.; Lim, S. N.; and Chellappa, R. 2018. Learning from synthetic data: Addressing domain shift for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3752–3761.

Taghanaki, S. A.; Abhishek, K.; Cohen, J. P.; Cohen-Adad, J.; and Hamarneh, G. 2021. Deep semantic segmentation of natural and medical images: A review. *Artificial Intelligence Review*, 54(1): 137–178.

Tobin, J.; Fong, R.; Ray, A.; Schneider, J.; Zaremba, W.; and Abbeel, P. 2017. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, 23–30. IEEE.

Treml, M.; Arjona-Medina, J.; Unterthiner, T.; Durgesh, R.; Friedmann, F.; Schuberth, P.; Mayr, A.; Heusel, M.; Hofmarcher, M.; Widrich, M.; et al. 2016. Speeding up semantic segmentation for autonomous driving. In *MLITS, NIPS Workshop*, volume 2.

Tsai, Y.-H.; Hung, W.-C.; Schulter, S.; Sohn, K.; Yang, M.-H.; and Chandraker, M. 2018. Learning to Adapt Structured Output Space for Semantic Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Vu, T.-H.; Jain, H.; Bucher, M.; Cord, M.; and Pérez, P. 2019. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2517–2526.

Wang, K.; Liew, J. H.; Zou, Y.; Zhou, D.; and Feng, J. 2019. Panet: Few-shot image semantic segmentation with prototype alignment. In *IEEE International Conference on Computer Vision (ICCV)*, 9197–9206.

Yang, H.; Zhang, R.; Guo, X.; Liu, W.; Zuo, W.; and Luo, P. 2020. Towards photo-realistic virtual try-on by adaptively generating-preserving image content. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7850–7859.

Yang, Y.; and Soatto, S. 2020. Fda: Fourier domain adaptation for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4085–4095.

Yue, X.; Zhang, Y.; Zhao, S.; Sangiovanni-Vincentelli, A.; Keutzer, K.; and Gong, B. 2019. Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. In *IEEE International Conference on Computer Vision (ICCV)*, 2100–2110.

Zhang, P.; Zhang, B.; Zhang, T.; Chen, D.; Wang, Y.; and Wen, F. 2021. Prototypical Pseudo Label Denoising and Target Structure Learning for Domain Adaptive Semantic Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhang, Q.; Zhang, J.; Liu, W.; and Tao, D. 2019. Category Anchor-Guided Unsupervised Domain Adaptation for Semantic Segmentation. In *Advances in Neural Information Processing Systems*, 433–443.

Zhang, Y.; David, P.; and Gong, B. 2017. Curriculum domain adaptation for semantic segmentation of urban scenes. In *IEEE International Conference on Computer Vision (ICCV)*, 2020–2030.

Zhou, K.; Yang, Y.; Qiao, Y.; and Xiang, T. 2021. Domain Generalization with MixStyle. In *International Conference on Learning Representations*.

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In *IEEE International Conference on Computer Vision (ICCV)*.

Zou, Y.; Yu, Z.; Kumar, B.; and Wang, J. 2018. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *European Conference on Computer Vision (ECCV)*, 289–305.