DIFFERENTIAL TREATMENT FOR STUFF AND THINGS: A SIMPLE UNSUPERVISED DOMAIN ADAPTATION METHOD FOR SEMANTIC SEGMENTATION

BY

ZHONGHAO WANG

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2019

Urbana, Illinois

Advisers:

Professor Thomas S. Huang
Adjunct Research Assistant Professor Honghui Shi

ABSTRACT

We consider the problem of unsupervised domain adaptation for semantic segmentation by easing the domain shift between the source domain (synthetic data) and the target domain (real data) in this work. State-of-the-art approaches prove that performing semantic-level alignment is helpful in tackling the domain shift issue. Based on the observation that stuff categories usually share similar appearances across images of different domains while things (i.e. object instances) have much larger differences, we propose to improve the semantic-level alignment with different strategies for stuff regions and for things: (1) for the **stuff** categories, we generate the feature representation for each class and conduct the alignment operation from the target domain to the source domain; (2) for the **thing** categories, we generate the feature representation for each individual instance and encourage the instance in the target domain to align with the most similar one in the source domain. In this way, the individual differences within thing categories will also be considered to alleviate over-alignment. In addition to our proposed method, we further reveal the reason why the current adversarial loss is often unstable in minimizing the distribution discrepancy and show that our method can help ease this issue by minimizing the most similar stuff and instance features between the source and the target domains. We conduct extensive experiments in two unsupervised domain adaptation tasks, GTA5 $\rightarrow$ Cityscapes and SYNTHIA $\rightarrow$ Cityscapes, and achieve the new state-of-the-art segmentation accuracy.

To my parents, for their love and support.

ACKNOWLEDGMENTS

I gratefully acknowledge the help and support of my advisors Thomas S. Huang and Honghui Shi throughout my researches. Their vision and guidance have helped me build the ability to pursue productive research.

I am grateful for the help offered by the Image Formation and Processing Group members. They are truly a great family to me.

I am thankful for my mother and father, who console me when I am depressed and cheer for me when I make an accomplishment.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

CNN	Convolutional Neural Network
GAN	Generative Adversarial Network
GPU	Graphics Processing Unit
mIoU	mean Intersection of Unions
SIM	Stuff and Instances Matching

CHAPTER 1

INTRODUCTION

Semantic segmentation [2] enables image scene understanding at the pixel level, which is crucial to many real-world applications such as autonomous driving. The recent surge of deep learning [3] methods that generate features from large training datasets has significantly accelerated the progress in semantic segmentation [4, 5, 6, 7]. However, collecting data with pixel-level annotations is costly in terms of both time and money. Specifically, to annotate an image in the widely used benchmark Cityscapes [8] dataset takes 1.5 hours on average; that sums up to 7,500 hours in total for annotating all the 5,000 images. Such annotation cost is quite burdensome, given that training deep neural networks on the collected data usually takes less than dozens of hours.

To address the problem of high-cost annotations, unsupervised domain adaptation methods are proposed for semantic segmentation [9, 10]. In these works, a model trained on a source domain dataset with segmentation annotations is adapted for an unlabeled target domain. The source domain datasets can be synthetic, e.g., from video games, so that little human effort is required. However, such methods suffer from the domain shift problem. Existing methods deal with the problem by minimizing the distribution discrepancy of the features extracted by a feature extractor [11, 12] between the source domain and the target domain. To this end, the GAN [13] structures, usually composed of a generator and a discriminator, are broadly used in this context. The generator extracts features from the input images, and the discriminator distinguishes which domain the features are generated from. The discriminator can thereby guide the generator to generate the target domain features with a distribution closer to the feature distribution of the source domain in an adversarial way.

In the previous GAN-style approaches, the adversarial loss is essentially a binary cross-entropy about whether the generated feature is from the source domain. We observe that such a global training signal is usually weak for the segmentation task. First, the alignments between stuff regions and between things require

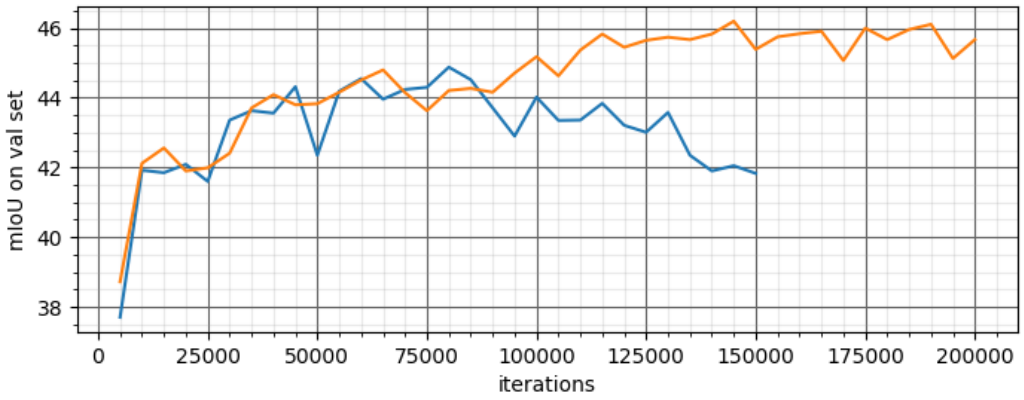

Figure 1.1: mIoU comparison on the validation set of Cityscapes by adapting from GTA5 dataset to Cityscapes dataset. The blue line corresponds to the output space adversarial adaptation strategy [1]. The red line corresponds to the output space adversarial adaptation combined with our proposed SIM structure. The model performance is tested every 5000 iterations.

different treatments but the adversarial loss lacks such structural information. For example, the stuff regions usually lack the appearance variance in an image but the things can have diverse appearances in the same image. Therefore, it is sub-optimal to use an adversarial loss to align the stuff and thing features globally without differential treatments. Second, the global GAN structure only adapts the feature distribution between two domains and does not necessarily adapt the target domain features towards the most likely space of source domain features. Therefore, as the semantic head gathers the features from the source domain with more training iterations, it becomes harder for the feature generator to adapt the target domain features exactly toward the source domain features. This leads to a performance drop on the target domain images as shown in figure 1.1.

This thesis proposes a stuff and instance matching (SIM) framework to address the aforementioned difficulties. First, we treat the alignments between stuff regions and between instances of things with different guidance. The key idea is shown in figure 1.2. The multiple stuff regions in a source image are usually similar, so the stuff from different domains can be directly aligned with their global feature vectors, while the multiple instances of the same thing, e.g., of the car category, can be diverse in the source image. Therefore we align instances in the target image to the most similar ones in the source image.

Second, we deal with the instability with the GAN training framework. We apply a L1 loss to explicitly minimize the distance between the target domain

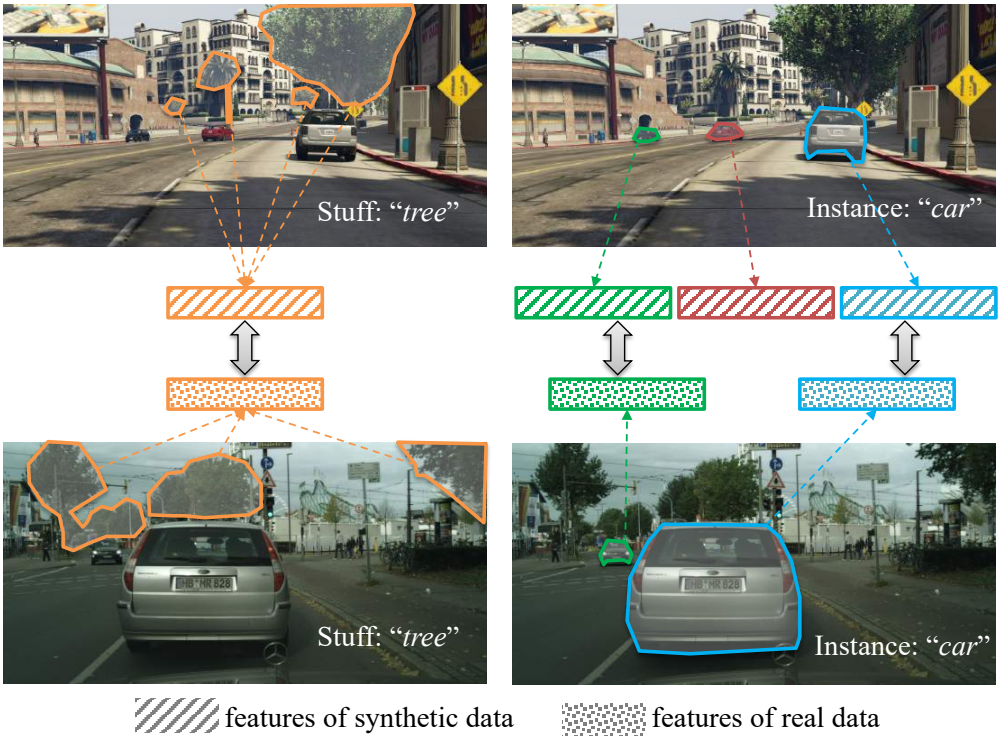

Figure 1.2: Illustration of the proposed and Stuff Instances Matching (SIM) structure. By matching the most similar stuff regions and things (i.e., instances) with differential treatments, we can adapt the features more accurately from the source domain to the target domain.

stuff and thing features with the most similar source domain counterparts. In this way, the adaptation is processed in a more accurate direction, instead of the rough distribution matching when using only the adversarial cross entropy loss, even after the semantic head gathers the source domain features with longer training iterations. As shown in figure 1.1, we implement the output space adversarial adaptation [1] from GTA5 [9] dataset to Cityscapes [8] dataset, and compare it with our model which adds the SIM module. We successfully solve the problem of the performance drop at longer training iterations with a few more computations.

Finally, we propose to improve the SIM framework with a self-supervised learning strategy. Specifically, we use predicted segmentation with high confidence to train the segmentation model, and to enhance the alignment for both stuff categories and thing categories.

We evaluate the proposed approach on two unsupervised domain adaptation tasks, the adaptation from GTA5 to Cityscapes and that from SYNTHIA to Cityscapes, and achieve a new state-of-the-art performance on both tasks.

CHAPTER 2

RELATED WORK

The domain adaptation in classification is a broadly studied problem after the surge of deep learning methods, and great progress has been made [14]. However, the domain adaptation in the semantic segmentation problem is more challenging as it is in essence a pixel-level classification problem involving structured contextual semantic adaptation. A typical practice of this task is adapting a semantic segmentation model trained on synthetic datasets [9, 10] (source domain) to perform on real image datasets [8] (target domain). The key idea of the domain adaptation task is to align the feature distributions between the source domain and the target domain, so that the model can utilize the knowledge learned from the source domain to perform tasks on the target domain. We generally divide current methods into three categories: image-level transferring, feature-level transferring and label-level transferring.

The image-level transferring refers to changing the appearance of images such that images from the source domain and the target domain are more visually similar. These methods [15, 16, 17] usually transfer the color, illumination and other stylization factors of images from one domain to another or from both domains to a neutral domain. In [15], Li et al. use CycleGAN [18] with a perceptual loss to preserve the locality of semantic information to perform the unpaired image-to-image transferring. In [17], Zhang et al. propose an Appearance Adaptation Network which transfers appearances of images between two domains mutually, such that the images' appearance tend to be domain-invariant. Choi et al. [19] propose a GAN-based self-ensembling data augmentation method for domain alignment.

The feature-level transferring refers to matching the extracted feature distributions between the source domain and the target domain. While feature extractors [11, 12, 20] can extract task-specific features, the features extracted from the target domain and those from the source domain have a discrepancy due to the domain shift, which negatively impacts the model's performance on the target domain dataset. Therefore, minimizing the feature distribution discrepancy with GAN

[13] structure is a common practice in domain adaptation. Sankaranarayanan et al. propose an image reconstruction framework [21] to make the reconstructed images from two domains close to each other so that the features are pulled closer with back propagation. Tsai and et al. propose a simple end-to-end output space domain adaptation framework [1]. Wu and et al. propose a channel-wise feature alignment network [16] to close the gap of the channel-wise mean and standard deviation in CNN feature maps. Chang and et al. propose a framework [22] to extract domain-invariant structures for adaptation.

The label-level transferring refers to giving pseudo-labels to the target domain dataset given the knowledge learned from the source domain for helping the adaptation task. This follows a self-supervised learning framework [23] where no human efforts are input for labeling the target dataset. Zou et al. [24] propose a class-balanced self-training framework. Li et al. [15] propose a joint self-learning and image transferring framework for adaptation.

CHAPTER 3

BACKGROUND

Definitions We follow the unsupervised semantic segmentation framework for the domain adaptation task; that is, given a source domain dataset with images and the pixel-level semantic annotations $\{x_i^s, y_i^s\}$ and a target domain dataset with only images $\{x_i^t\}$, we plan to train a model that can predict the pixel-level labels $\{\hat{y}_i^t\}$ for the target domain images. We denote the class indices set with N.

Segmentation and adversarial adaptation The semantic segmentation task in deep learning literature is broadly discussed [4, 5, 6, 7], and the problem-solving strategy is formalized by utilizing a feature extractor network F to extract image features and a classification head C to classify features into semantic classes. We use the cross entropy loss to supervise the model on the pixel classification task with the annotated source domain dataset in Eqn (3.1).

$$\mathfrak{L}_{seg}^{S}(f_i^s) = -\sum_{i,h,w}\sum_{k\in N} y_i^{s(h,w)} \log(\mathcal{S}(C(f_i^s)^{(h,w)})^{(k)}) \tag{3.1}$$

where $f_i^s = F(x_i^s)$, $x_i^s \in X^s$, X^s is the source domain image dataset, h and w are the height index and the width index of the feature maps, y is the ground truth label, $\mathcal{S}$ is the softmax operation. However, due to the domain shift problem, the model trained on the source domain will achieve inferior performance if directly applied to test on the target domain. Therefore, we impose a traditional GAN structure on the output space [1] to globally minimize the feature distribution discrepancy between the source domain and the target domain. Here, the feature extractor F and the classification head C serve as the generator G where $G = C \circ F$. A discriminator D will discriminate the output of G. We close the feature distribution discrepancy between the source domain and the target domain by optimizing the adversarial target function in Eqn (3.2).

$$\min_{G} \mathfrak{L}_{adv}(G, D) = - \sum_{x_i^t \in X^T} \log(1 - D(\mathcal{S}(G(x_i^t)))) \tag{3.2}$$

while the discriminator tries to distinguish which domain the feature is from by optimizing the discriminator target function in Eqn (3.3).

$$\begin{aligned} \min_{D} \mathfrak{L}_{D}(G, D) = & - \sum_{x_i^t \in X^T} \log(D(\mathcal{S}(G(x_i^t)))) \\ & - \sum_{x_i^s \in X^S} \log(1 - D(\mathcal{S}(G(x_i^s)))) \end{aligned} \tag{3.3}$$

CHAPTER 4

STUFF AND INSTANCES MATCHING FRAMEWORK

The key idea of our method is that the past experience leading to good outcomes should also help the current training process. Specifically to our task, the past experience should help both the feature-level transferring and the label-level transferring from the source domain to the target domain. First, we raise a stuff and instance matching (SIM) framework to reduce the intra-class domain shift problem. Second, we propose a self-supervised learning framework combined with our proposed SIM structure to enable the label-level transferring, which further boosts the performance. The overall framework is shown in figure 4.1.

4.1 Stuff and instance matching (SIM)

First, we discuss the matching process for the background classes such as road, sidewalk, sky etc.. These classes usually cover a large area of the image and lack appearance variation, so we only extract the image-level stuff feature representation for them. For each source domain image, we access the correctly classified label map by selecting the predicted labels matched with the ground truth labels in Eqn (4.1).

$$\begin{aligned} L^s_{P_i} &= \underset{k \in N}{\operatorname{argmax}}(C(f^s_i)^{(k)}) \\ L^s_{C_i} &= L^s_{G_i} \cap L^s_{P_i} \end{aligned} \tag{4.1}$$

where $L^s_{C_i}$ is the correctly classified label map, $L^s_{G_i}$ is the ground truth label map, $L^s_{P_i}$ is the predicted label map, and $i \in \{1..|X^S|\}$. We average the features belonging to the same background semantic class across the width and height of the image as the stuff representation for each background class in Eqn (4.2).

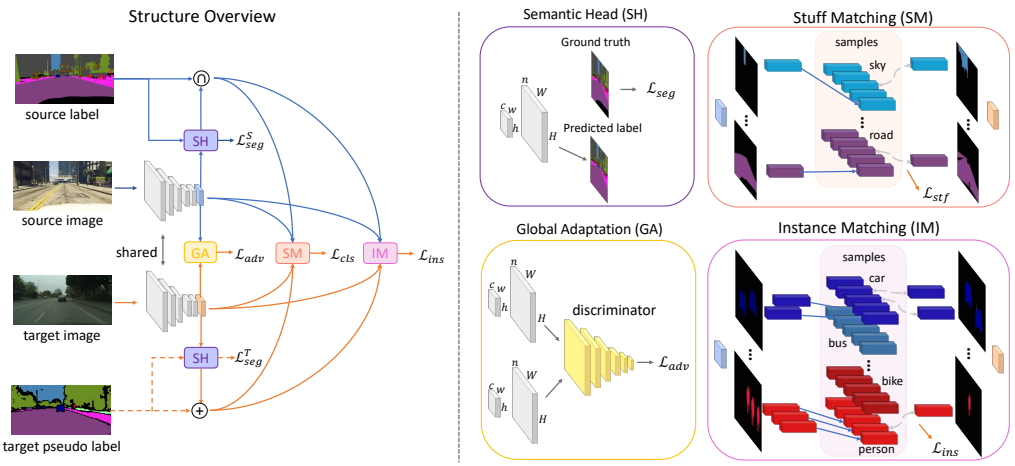

Figure 4.1: Framework. (1) The overall structure is shown on the left. The solid lines represent the first step training procedure in Eqn (4.9), and the dash lines along with the solid lines represent the second step training procedure in Eqn (4.10). The blue lines correspond to the flow direction of the source domain data, and the orange lines correspond to the flow direction of target domain data. $\cap$ is an operation defined in Eqn (4.1); $+$ is an operation defined in Eqn (4.8) and is only effective in the second step training procedure. (2) The specific module design is shown on the right. h, w and c represent the height, width and channels for the feature maps; H, W and n represent the height, width and class number for the output maps of the semantic head. For SH, the input ground truth label map supervises the the semantic segmentation task, and the semantic head also generates a predicted label map joining the operations of $\cap$ and $+$. For SM and IM, the grey dash lines represent the matching operation defined in Eqns (4.3) and (4.5) respectively.

$$\begin{aligned} \mathcal{A}^b(L, f) &= \frac{\sum_{h,w} \delta(L^{(h,w)} - b) f^{(h,w)}}{\max(\epsilon, \sum_{h,w} \delta(L^{(h,w)} - b))} \\ S_j^b &= \mathcal{A}^b(L_{C_i}^s, f_i^s) \quad \textbf{where } j = i \bmod w, \\ &\qquad\qquad \textbf{if } \mathcal{A}^b(L_{C_i}^s, f_i^s) \neq 0 \end{aligned} \tag{4.2}$$

where S_j^b is the j'th source domain semantic feature sample of class b, $b \in B$ (background classes), $i \in \{1..|X^S|\}$, w is the number of feature samples to be stored for each class, δ is the Dirac delta function and ϵ is a regularizing term. For each target domain image, we minimize the distance of the stuff representation of each background class with the closest intra-class source stuff feature representation. Because the ground truth of the target domain image is not provided, we use the predicted label map to generate the stuff feature representation for each background class. We adapt the stuff feature representation of the background classes

by minimizing the loss function defined in Eqn (4.3) when the model is trained on the target domain.

$$\mathfrak{L}_{stf} = \sum_{i} \sum_{b} \min_{j} \left\| \mathcal{A}^{b}(L^{t}_{P_i}, f^{t}_{i}) - S^{b}_{j} \right\|^{1}_{1} \tag{4.3}$$

where $i \in \{1..|X^T|\}$, and $b \in L^{t}_{P_i} \cap B$.

Second, we discuss the instance matching process for the foreground classes such as cars, persons etc.. Because the ground truth does not provide the instance level annotations, we generate the foreground instance mask by finding the disconnected regions for each foreground class in the label map L. This coarsely segments the intra-class semantic regions into multiple instances, and thus various instance-level feature representations of one image can be generated accordingly in Eqn (4.4).

$$\begin{aligned} R_k &= \{r_{k_1}, r_{k_2}, ..., r_{k_m}\} = \mathcal{T}(L, k) \\ \mathcal{I}(r, f) &= \frac{\sum_{h,w} r^{(h,w)} f^{(h,w)}}{\max(\epsilon, \sum_{h,w} r^{(h,w)})} \end{aligned} \tag{4.4}$$

where r_{k_i} is the i'th ($i \in \{1, .., m\}$) binary mask of the connected region belonging to class k, $k \in K$ (foreground classes), $\mathcal{T}$ is the operation to find the disconnected regions of class k from the label mask L, and $\mathcal{I}$ is the operation to generate the instance-level feature representation. The source domain instance feature samples can be generated in algorithm 1. Therefore, the target domain instance features can be pulled closer to the closest intra-class source domain instance feature sample by minimizing the loss function in Eqn (4.5).

$$\mathfrak{L}_{ins} = \sum_{i} \sum_{k \in K} \frac{1}{\left|R^{t}_{k}\right|} \sum_{r^t \in R^{t}_{k}} \min_{j} \left\| \mathcal{I}(r^t, f^{t}_{i}) - S^{k}_{j} \right\|^{1}_{1} \tag{4.5}$$

where $i \in \{1..|X^T|\}$, and $R^{t}_{k} = \mathcal{T}(L^{t}_{P_i}, k)$.

4.2 Self-supervised learning with SIM

Because the model is only trained on the source domain with the ground truth annotations, the features and the softmax output are thus generated to optimize the source domain segmentation loss function but ignore the target domain segmen-

Algorithm 1: Instance-level source feature samples

Result: S^k
$z = 10$; *# maximum class instances in an image*
$c_k = 0, \forall k \in K$; *# instance feature counter*
for $x_i^s \in X^S$ **do**
  for $k \in K$ **do**
    $R_k^s = \mathcal{T}(L_{C_i}^s, k)$
    if $R_k^s \neq \emptyset$ **then**
      R_{sort} = sort R_k^s by area in descent order
      for $l \in \{1..\min(z, |R_{sort}|)\}$ **do**
        $j = c_k \mod z * w$
        $c_k = c_k + 1$
        $S_j^k = \mathcal{I}(R_{sort}[l], f_i^s)$
      end
    end
  end
end

tation supervision. However, the distribution of the ground truth labels from both domains also have a discrepancy, and this negatively impacts the model's performance on the target domain. Therefore, we propose a self-supervised learning framework combined with our feature matching methods to alleviate this problem.

We first follow the framework described in chapter 3 and section 4.1 to train a model with the source domain images X^S and ground truth annotations Y^S along with the target domain images X^T. Then we use the trained model to give pseudo-labels to the pixels with high confidence of the predicted labels in the training set images X^T shown in Eqn (4.6).

$$\hat{y}_i^t = \underset{k \in N}{\operatorname{argmax}} \, \mathbb{1}_{[\mathcal{S}(C(f_i^t))^{(k)} > y_t^k]}(C(f_i^t)^{(k)}) \tag{4.6}$$

where $\mathbb{1}$ is a function which returns the input if the condition is true or a *don't care* symbol if not, and y_t^k is the confidence threshold for class k. Then, we add the semantic segmentation loss on the target domain images in Eqn (4.7) along with other losses to retrain our model.

$$\mathfrak{L}_{seg}^T(f^t) = -\sum_{i,h,w} \sum_{k \in N} \hat{y}_i^{(h,w)} \log(\mathcal{S}(C(f_i^t)^{(h,w)})^{(k)}) \tag{4.7}$$

With the pseudo labels supervising the model to generate features corresponding to specific classes, these features should generically be adapted to be closer to the

corresponding intra-class source domain features. The $L^t_{P_i}$ is thereby augmented by Eqn (4.8) for the stuff feature adaptation loss defined in Eqn (4.3) and the instance feature adaptation loss defined in Eqn (4.5):

$$\mathbb{1}_{L^t_{P_i} \neq \hat{y}^t_i}(L^t_{P_i}) = \mathbb{1}_{L^t_{P_i} \neq \hat{y}^t_i}(\hat{y}^t_i) \tag{4.8}$$

$\mathbb{1}$ selects the positions in the input satisfying the condition.

4.3 Training procedure

We follow a two-step training procedure to improve the performance of the generator G on semantic segmentation task on the target domain dataset. First, we train our model without the self-supervised learning module, and optimize the target function in Eqn (4.9) with G and D in an adversarial training strategy:

$$\begin{aligned} \min_{G,D} \mathfrak{L}_{step1} = &\min_{G}(\lambda_{seg}\mathfrak{L}^S_{seg} + \lambda_{adv}\mathfrak{L}_{adv} + \\ &\lambda_{ci}(\mathfrak{L}_{stf} + \mathfrak{L}_{ins})) + \min_{D} \lambda_D \mathfrak{L}_D \end{aligned} \tag{4.9}$$

where λ's are the weight parameters for the losses. Second, after giving the pseudo labels to the target domain training dataset with the model trained in the first step, we reinitialize and repeat the training process to optimize the loss function in Eqn (4.10).

$$\begin{aligned} \min_{G,D} \mathfrak{L}_{step2} = &\min_{G}(\lambda_{seg}(\mathfrak{L}^S_{seg} + \mathfrak{L}^T_{seg}) + \lambda_{adv}\mathfrak{L}_{adv} + \\ &\lambda_{ci}(\tilde{\mathfrak{L}}_{stf} + \tilde{\mathfrak{L}}_{ins})) + \min_{D} \lambda_D \mathfrak{L}_D \end{aligned} \tag{4.10}$$

where $\tilde{\mathfrak{L}}_{stf}$ and $\tilde{\mathfrak{L}}_{ins}$ are augmented with predicted $\hat{y}^t_i$s according to Eqn (4.8).

CHAPTER 5

IMPLEMENTATIONS

5.1 Network architecture

Segmentation Network. We adopt ResNet-101 model [12] pre-trained on ImageNet [25] with only the 5 convolutional layers $\{conv1, res2, res3, res4, res5\}$ as the backbone network. Due to memory limit, we do not use the multi-scale fusion strategy [26]. For generating better-quality feature maps, we follow the common practice from [4, 26, 1] and twice the resolution of the feature maps of the final two layers. To enlarge the field of view, we use dilated convolutional layers [26] with stride 2 and 4 in $res4$ and $res5$. For the classification heads, we apply an ASPP module [5] to $res5$ with $\lambda_{seg} = 1$.

Discriminator. Following [1], we use 5 convolutional layers with kernel size 4x4, stride 2 and channel numbers $\{64, 128, 256, 512, 1\}$ to form the network. We use a leaky ReLU [27] layer of -0.2 slope between adjacent convolutional layers. Due to the small batch size in the training process, we do not use batch normalization layers [28]. The sole discriminator is implemented on the upsampled softmax output of the ASPP head on $res5$ with $\lambda_{adv} = 0.001$ and $\lambda_D = 1$.

5.2 Training details

We use Pytorch toolbox and a single GPU to train our network. Stochastic gradient descent (SGD) is used to optimize the segmentation network. We use Nesterov's method [29] with momentum 0.9 and weight decay 5×10^{-4} to accelerate the convergence. Following [4], we set the initial learning rate to be 2.5×10^{-4} and let it polynomially decay with the power of 0.9. For the discriminator networks, we use Adam optimizer [30] with momentum 0.9 and 0.99. The initial learning rate is set to 10^{-4} and the same polynomial decay rule is applied.

CHAPTER 6

EXPERIMENTS

6.1 Datasets

The Cityscapes [8] dataset consists of 5000 images of resolution 2048×1024 with high-quality pixel-level annotations. These images of street scenes were annotated with 19 semantic labels for evaluation. This dataset is split into training, validation and test sets with 2975, 500 and 1525 images respectively. Following previous works [31, 32], we only evaluate our models on the validation set. The GTA5 [9] dataset contains 24966 fine annotated synthetic images of resolution 1914×1052. All the images are frames captured from the game Grand Theft Auto V. To accommodate the model with the limited GPU memory, we follow [1] and resize GTA5 images to the resolution of 1280×720. This dataset shares all the 19 classes used for evaluation in common with the Cityscapes dataset. The SYNTHIA [10] dataset has 9400 images of resolution 1280×760 with pixel-level annotations. Similar to [33, 1, 34, 15], we evaluate our models on Cityscapes validation set with the 13 classes shared in common between SYNTHIA dataset and Cityscapes dataset. The Cityscapes images are resized to 1024×512 for both the training stage and the testing stage.

6.2 GTA5 to Cityscapes

We first present our overall results and compare to the previous state-of-the-art methods; then we discuss the effectiveness of each module in our model; finally we discuss the choice of hyperparameters of our proposed SIM module.

Overall results. We compare the performance of our method with the current state-of-the-art methods in table 6.1. For fair comparison, we list the performance of the models using resnet-101 [12] as the backbone. Our method achieves a new

Table 6.1: Comparison to the state-of-the-art results of adapting GTA5 to Cityscapes.

GTA5 → Cityscapes																				
Method	road	sidewalk	building	wall	fence	pole	light	sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorbike	bike	mIoU
Wu et al.[35]	85.0	30.8	81.3	25.8	21.2	22.2	25.4	26.6	83.4	36.7	76.2	58.9	24.9	80.7	29.5	42.9	2.5	26.9	11.6	41.7
Tsai et al.[1]	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
Saleh et al.[36]	79.8	29.3	77.8	24.2	21.6	6.9	23.5	44.2	80.5	38.0	76.2	52.7	22.2	83.0	32.3	41.3	**27.0**	19.3	27.7	42.5
Luo et al. [33]	88.5	35.4	79.5	26.3	24.3	28.5	32.5	18.3	81.2	40.0	76.5	58.1	25.8	82.6	30.3	34.4	3.4	21.6	21.5	42.6
Hong et al.[37]	89.2	**49.0**	70.7	13.5	10.9	38.5	29.4	33.7	77.9	37.6	65.8	**75.1**	32.4	77.8	39.2	45.2	0.0	25.5	35.4	44.5
Chang et al. [22]	**91.5**	47.5	82.5	31.3	25.6	33.0	33.7	25.8	82.7	28.8	82.7	62.4	30.8	85.2	27.7	34.5	6.4	25.2	24.4	45.4
Du et al. [34]	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
Vu et al. [38]	89.4	33.1	81.0	26.6	26.8	27.2	33.5	24.7	83.9	36.7	78.8	58.7	30.5	84.8	38.5	44.5	1.7	31.6	32.4	45.5
Chen et al. [39]	89.4	43.0	82.1	30.5	21.3	30.3	34.7	24.0	85.3	39.4	78.2	63.0	22.9	84.6	36.4	43.0	5.5	**34.7**	33.5	46.4
Zou et al. [24]	89.6	58.9	78.5	33.0	22.3	**41.4**	**48.2**	**39.2**	83.6	24.3	65.4	49.3	20.2	83.3	39.0	48.6	12.5	20.3	35.3	47.0
Lian et al. [40]	90.5	36.3	84.4	32.4	**28.7**	34.6	36.4	31.5	**86.8**	37.9	78.5	62.3	21.5	**85.6**	27.9	34.8	18.0	22.9	**49.3**	47.4
Li et al. [15]	91.0	44.7	84.2	**34.6**	27.6	30.2	36.0	36.0	85.0	**43.6**	83.0	58.6	**31.6**	83.3	35.3	**49.7**	3.3	28.8	35.6	48.5
ours	90.6	44.7	**84.8**	34.3	**28.7**	31.6	35.0	37.6	84.7	43.3	**85.3**	57.0	31.5	83.8	**42.6**	48.5	1.9	30.4	39.0	**49.2**

Table 6.2: Ablation study on the adaptation from GTA5 dataset to Cityscapes dataset. AA stands for adversarial adaptation; IT stands for image transferring; SIM stands for semantic and instance matching; SSL stands for self-supervised learning.

method	AA	IT	SIM	SSL	mIoU
source only					36.6
+ AA[1]	✓				41.4
+ IT[15]	✓	✓			44.9
+ SIM	✓	✓	✓		46.2
+ SSL	✓	✓	✓	✓	49.2
target only					65.1

state of the art.

Module contributions. We show the contribution of each module to the overall performance of our model in table 6.2. If trained purely on the source domain dataset, the model can achieve an mIoU of 36.6 on the Cityscapes validation set. Then, we follow the work of [1] to add the global adversarial training on the output space with the adversarial loss in Eqn (3.2) and the discriminator loss in Eqn (3.3), and the mIoU is thereby improved to 41.4. As mentioned in Chapter 2, image-level adaptation is also a key factor in minimizing the discrepancy of data distribution. Therefore, it is helpful to utilize a transferred source-domain image dataset whose appearance is more similar to that of the target-domain image dataset. We adopt the transferred GTA5 images of [15] which utilizes a CycleGAN [18] structure to adapt the style of GTA5 images to the style of Cityscapes

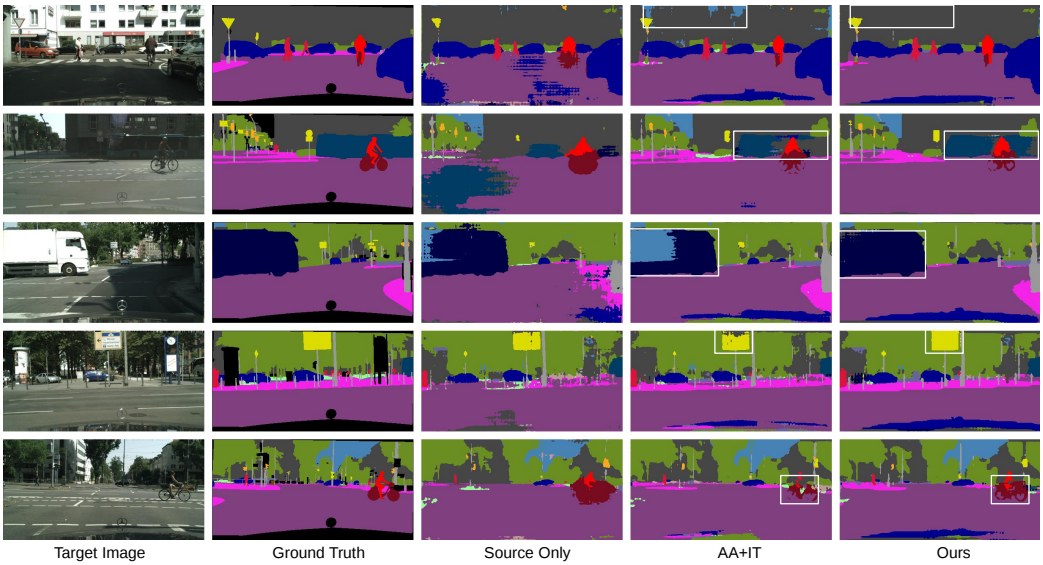

Figure 6.1: Visualization of the segmentation results. Source only, AA+IT, and Ours correspond to the models that achieve mIoU of 36.6, 44.9, and 49.2 in table 6.2, respectively.

images. This further improves the mIoU to 44.9, which serves as the baseline for our work.

Then, we add our SIM module to the training framework. The background classes include road, sidewalk, building, wall, fence, vegetation, terrain and sky. The foreground classes are all the remaining classes used for evaluation. With the best setting for the SIM module where $\lambda_{ci} = 0.01$ and w, the number of semantic source domain feature samples to be stored is 50 and the mIoU improves to 46.2 by optimizing the Eqn (4.9). In this setting, we empirically set the maximum source domain instance features of each class to be stored to 10 for each image, and the feature of the instance covering larger area is to be stored with higher priority. We also adapt 10 instance features at maximum for each class from the target domain to the source domain. This is because instance feature representations of small regions or noise regions may be too numerous for storage and adaptation. For example, there are many dots corresponding to the bike class in the image at the intersection of the second row and last column in figure 6.1; all these dots are segmented into separate regions, and it would be inefficient to adapt all of them from the target domain to the source domain.

Finally, we retrain our model with the combination of SIM and the self-supervised learning (SSL) framework given the pseudo-labeled target dataset by the training

Table 6.3: Influence of λ_{ci} given the number of semantic feature samples to be stored is 50 ($w = 50$).

λ_{ci}	0.1	0.05	0.01	0.005	0.001
mIoU	43.4	44.2	**46.2**	45.4	45.5

Table 6.4: Influence of the number of semantic feature samples to be stored (w) given $\lambda_{ci} = 0.01$.

w	10	50	200	800	1600
mIoU	45.2	**46.2**	46.1	45.3	45.0

step 1. When generating the pseudo labels for the target dataset, we choose the confidence threshold for each class. We first follow Eqn (4.6) to give pseudo labels for each pixel by setting $y_t = 0$ for each image in the target dataset. Then, we generate a confidence map corresponding to the pseudo label map where the confidence is the maximum item of the softmax output in each channel so that the pseudo label at each pixel is associated with a confidence value. After this, we rank the confidence values belonging to the same class across the whole target dataset. If the median confidence value is below 0.9, then the confidence threshold for that class is set to the median confidence value; otherwise, it is set to 0.9 exactly. With the new y_t^k being set, we follow Eqn (4.6) to generate the pseudo labels with *don't cares* for the target dataset and thus the model retraining can be processed by optimizing the Eqn (4.10). This improves the mIoU to 49.2. Furthermore, we compare our model with the oracle model [1] trained on the target dataset without any transferring method. There is a gap of 15.9%, indicating that further studies on this problem are necessary. We provide a visualization showing the improvements of our methods in figure 6.1.

Hyperparameters analysis. This mainly deals with the settings of λ_{ci}, the weight for the semantic matching loss and the instance matching loss, and w, the number of semantic feature samples to be stored for our proposed SIM module. For the hyperparameters of other modules, we follow [1] to set $\lambda_{seg} = 1$, $\lambda_{adv} = 0.01$ and $\lambda_D = 1$ to control the variables.

First, we discuss the influence of λ_{ci} given $w = 50$, which is shown in table 6.3. We test the influence of λ_{ci} with different w's. Here we only exhibit the results with $w = 50$, the setting that achieves the best performance, to provide intuition into the influence of the choice of λ_{ci}. We argue that λ_{ci} should not

Table 6.5: Comparison to the state-of-the-art results of adapting SYNTHIA to Cityscapes.

SYNTHIA → Cityscapes														
Method	road	sidewalk	building	light	sign	vegetation	sky	person	rider	car	bus	motorbike	bike	mIoU
Luo et al. [33]	82.5	24.0	79.4	16.5	12.7	79.2	82.8	58.3	18.0	**79.3**	25.3	17.6	25.9	46.3
Tsai et al.[1]	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
Du et al. [34]	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
Li et al. [15]	**86.0**	**46.7**	80.3	14.1	11.6	79.2	81.3	54.1	27.9	73.7	**42.2**	25.7	45.3	51.4
ours	83.0	44.0	80.3	**17.1**	**15.8**	80.5	81.8	**59.9**	**33.1**	70.2	37.3	**28.5**	**45.8**	**52.1**

be set either too large or too small. If it is too large, the features corresponding to the image-level or instance-level semantic class would be pulled too close to the same source domain feature sample, such that these target-domain features would also be very close to each other and thus would lack intra-class feature variance. This could worsen the scene understanding for the feature extractor and thus negatively impact the overall performance of our model. On the other hand, if λ_{ci} is too small, the matching loss would not help the model much in minimizing the feature discrepancy between the source domain and the target domain. As shown in table 6.3, when $\lambda_{ci} = 0.01$, an appropriately large value, the model achieves the best performance.

Second, we show the influence of the choice of w, the number of semantic feature samples to be stored, in table 6.4. As the model is always being updated during the training stage, it would be infeasible to access all the source-domain feature samples with the newly updated model. Therefore, we store a number of feature samples generated with recent updated models. The number of these feature samples, w, should balance the factors such that (1) w should be large enough so that there will be enough source domain feature samples to be matched; and (2) w should not be so large that the stored source domain feature samples are not up-to-date. With our experiments, $w = 50$ achieves the best performance.

6.3 SYNTHIA to Cityscapes

We evaluate the mIoU of 13 classes shared between the source domain and the target domain as [33, 1, 34, 15]. We use the same hyperparameters which achieve

Table 6.6: Ablation study on the adaptation from SYNTHIA dataset to Cityscapes dataset. AA stands for adversarial adaptation; IT stands for image transferring; SIM stands for semantic and instance matching; SSL stands for self-supervised learning.

method	AA	IT	SIM	SSL	mIoU
source only					38.6
+ AA[1]	✓				45.9
+ IT[15]	✓	✓			46.0
+ SIM	✓	✓	✓		47.1
+ SSL	✓	✓	✓	✓	52.1
target only					71.7

the best performance discussed in section 6.2 for all the following experiments. We compare our model with the previous state-of-the-art models in table 6.5. Our model also achieves a new state of the art on adaptation from SYNTHIA dataset to the Cityscapes dataset.

Table 6.6 shows the contribution of each module. The model can achieve an mIoU of 38.6 if trained on the source domain only. By adding the adversarial training module and utilizing the transferred source domain images, the model can achieve an mIoU of 46.0. We notice that the improvement of utilizing the transferred images is not obvious, and we conjecture that this is because of the large gap between the layouts of the source domain and the target domain. By adding our SIM module, the mIoU improves to 47.1. After retraining our model with self-supervised learning using the same pseudo-labeling strategy described in section 6.2, our model achieves an mIoU of 52.1.

CHAPTER 7

CONCLUSIONS

We propose a stuff and instance matching (SIM) module for the unsupervised domain adaptation of semantic segmentation from a synthetic dataset to a real-image dataset. We (1) consider the difference of appearance variance between the stuff regions and the instances of things, and thus treat them differently in the adaptation process; (2) explicitly minimize the distance of the closest stuff and instance features between the source domain and the target domain, which enables the adaptation in a more accurate direction and stabilizes the GAN training process at longer iterations. When combined with self-training, our SIM model achieves a new state of the art on this task.

REFERENCES

[1] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker, "Learning to adapt structured output space for semantic segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[2] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature Cell Biology*, vol. 521, no. 7553, pp. 436–444, 5 2015.

[4] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *arXiv preprint arXiv:1412.7062*, 2014.

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE TPAMI*, vol. 40, no. 4, pp. 834–848, 2017.

[6] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *IEEE CVPR*, 2017.

[7] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[8] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[9] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 102–118.

[10] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-Scale image recognition," *arXiv e-prints*, p. arXiv:1409.1556, Sep 2014.

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

[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680. [Online]. Available: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

[14] L. Zhang, "Transfer adaptation learning: A decade survey," *arXiv e-prints*, p. arXiv:1903.04687, Mar 2019.

[15] Y. Li, L. Yuan, and N. Vasconcelos, "Bidirectional learning for domain adaptation of semantic segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[16] Z. Wu, X. Han, Y.-L. Lin, M. Gokhan Uzunbas, T. Goldstein, S. Nam Lim, and L. S. Davis, "DCAN: Dual channel-wise alignment networks for unsupervised scene adaptation," in *The European Conference on Computer Vision (ECCV)*, September 2018.

[17] Y. Zhang, Z. Qiu, T. Yao, D. Liu, and T. Mei, "Fully convolutional adaptation networks for semantic segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[18] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[19] J. Choi, T. Kim, and C. Kim, "Self-ensembling with GAN-based data augmentation for domain adaptation in semantic segmentation," in *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[20] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[21] S. Sankaranarayanan, Y. Balaji, A. Jain, S. Nam Lim, and R. Chellappa, "Learning from synthetic data: Addressing domain shift for semantic segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[22] W.-L. Chang, H.-P. Wang, W.-H. Peng, and W.-C. Chiu, "All about structure: Adapting structural information across domains for boosting semantic segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[23] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *arXiv e-prints*, p. arXiv:1902.06162, Feb 2019.

[24] Y. Zou, Z. Yu, B. Vijaya Kumar, and J. Wang, "Unsupervised domain adaptation for semantic segmentation via class-balanced self-training," in *The European Conference on Computer Vision (ECCV)*, September 2018.

[25] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, June 2009, pp. 248–255.

[26] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *ICLR*, 2016.

[27] A. L. Maas, A. Y Hannun, and A. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *ICML*, 2013.

[28] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015. [Online]. Available: http://dl.acm.org/citation.cfm?id=3045118.3045167

[29] A. Botev, G. Lever, and D. Barber, "Nesterov's accelerated gradient and momentum as approximations to regularised update descent," in *IEEE IJCNN*, 2017, pp. 1899–1903.

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2014.

[31] J. Hoffman, D. Wang, F. Yu, and T. Darrell, "FCNs in the wild: Pixel-level adversarial and constraint-based adaptation," *arXiv preprint arXiv:1612.02649*, 2016.

[32] X. Peng, B. Usman, N. Kaushik, J. Hoffman, D. Wang, and K. Saenko, "VisDA: The Visual Domain Adaptation Challenge," *arXiv preprint arXiv:1710.06924*, 2017.

[33] Y. Luo, L. Zheng, T. Guan, J. Yu, and Y. Yang, "Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[34] L. Du, J. Tan, H. Yang, J. Feng, X. Xue, Q. Zheng, X. Ye, and X. Zhang, "SSF-DAN: Separated semantic feature based domain adaptation network for semantic segmentation," in *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[35] Z. Wu, X. Han, Y. Lin, M. G. Uzunbas, T. Goldstein, S. Lim, and L. S. Davis, "DCAN: Dual channel-wise alignment networks for unsupervised scene adaptation," in *ECCV*, 2018. [Online]. Available: http://arxiv.org/abs/1804.05827

[36] F. Saleh, S. Aliakbarian, M. Salzmann, L. Petersson, and J. M. Alvarez, "Effective use of synthetic data for urban scene semantic segmentation," in *ECCV*, 2018.

[37] W. Hong, Z. Wang, M. Yang, and J. Yuan, "Conditional generative adversarial network for structured domain adaptation," in *IEEE CVPR*, 2018.

[38] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Perez, "Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[39] M. Chen, H. Xue, and D. Cai, "Domain adaptation for semantic segmentation with maximum squares loss," in *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[40] Q. Lian, F. Lv, L. Duan, and B. Gong, "Constructing self-motivated pyramid curriculums for cross-domain semantic segmentation: A non-adversarial approach," in *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.