Semantic Segmentation and Adversarial Domain Adaptation

Yifan Bao
Zhejiang University

Rui Feng
Shanghai Jiao Tong University

Abstract

Deep neural networks for semantic segmentation always require a large number of samples with pixel-level ground truth, which becomes the major difficulty in unseen image domains. To reduce the intensive labor cost for labeling, unsupervised domain adaptation (UDA) approaches are proposed to adapt source ground truth labels to the target domain. In the report, we use an adversarial learning method for domain adaptation in semantic segmentation task. We adopt adversarial learning in the output space, considering semantic segmentation as structured outputs that contain spatial similarities between the source and target domains. We compare two different segmentation models and show that a better segmentation model has more significant performance improvement using adversarial training. We also construct a multi-level adversarial network to effectively perform output space domain adaptation at different feature levels. Both synthetic-to-real and between-city adaptation experiments demonstrate the effectiveness of our approach.

1 Introduction

Image segmentation is one of the most popular topics in computer vision with a lot of applications. It is the process of partitioning a digital image into multiple segments. Recently, deep learning-based (DL-based) image segmentation models, often achieving the highest accuracy, have received a lot of attention [1, 2, 3, 4]. Nonetheless, these methods typically require a huge amount of labeled data matching the considered scenario to obtain reliable performances. The collection and annotation of large datasets for every new task and domain are extremely expensive, time-consuming, and error-prone. Furthermore, in many scenarios, sufficient training data may not be available for other domains and tasks that are in some way related to the considered one. Hence, the ability to use a model trained on correlated samples from a different task would highly benefit real-world applications.

However, despite of the great success DL-based methods have achieved in segmentation area, these approaches are often suffered from generalization problem. Unsupervised domain adaptation (UDA) is the field of research that aims to tackle this problem. It designs to learn only from source domain (with labeled data) a well performing model on target domain (with unlabeled data). In semantic segmentation tasks where pixel-by-pixel labeling is required, samples annotation is the most demanding task, while data acquisition is much simpler and cheaper. For this reason, in this project, we will cover the UDA on semantic segmentation.

To be more specific, in this project, we conduct two major tasks: I. Try some semantic segmentation algorithms on different datasets and evaluate the results. II. Use unsupervised adversarial domain adaptation on semantic segmentation models to improve the generalization ability and robustness of different segmentation models.

Preprint. Under review.

2 Related Work

2.1 Segmentation Models

Long et al. [1] propose one of the first deep learning works for semantic image segmentation using a fully convolutional network (FCN). It replaces all fully connected layers with the fully convolutional layers and outputs a spatial segmentation map. Based on FCN, numerous methods have since been designed to improve this model by utilizing context information or enlarging receptive fields. DeepLabv1 (see [3]) combines Deep Convolutional Neural Networks (DCNN) with probabilistic graphical models in order to overcome the poor localization property of deep network. DeepLabv2 (see [4]) introduces atrous spatial pyramid pooling (ASPP) to segment objects at multiple scales and replaces VGG16 with ResNet as the backbone. DeepLabv3 (see [5]) and DeepLabv3+ (see [6]) are the latest models in DeepLab family. The former one has both cascaded and parallel modules of dilated convolutions, while the latter one uses the DeepLabv3 framework as encoder part and further adds a decoder module to obtain sharp object boundaries.

2.2 Domain Adaptation

Domain adaptation methods for image classification have been developed to address the domain-shift problem between the source and target domains. Numerous methods [7, 8, 9] are developed based on CNN classifiers due to performance gain. The main insight behind these approaches is to tackle the problem by aligning the feature distribution between source and target images. Ganin et al. [7, 8] propose the Domain-Adversarial Neural Network(DANN) to transfer the feature distribution. A number of variants have since been proposed with different loss functions [9]. The PixelDA method [10] addresses domain adaptation for image classification by transferring the source images to target domain, thereby obtaining a simulated training set for target images.

Hoffman et al. [11] introduce the task of domain adaptation on semantic segmentation by applying adversarial learning (i.e., DANN) in a fully-confolutional way on feature representations and additional category constraints similar to the constrained CNN [12]. Other methods focus on adapting synthetic-to-real or cross-city images by adopting class-wise adversarial learning [13] or label transfer.

Recently, output adversarial adaptation has drawn much attention. Tsai et al. [14] are the first to propose this type of adaptation: in order to improve the signal flow from the adversarial competition through the segmentation network, they deploy multiple dense classification modules at different depths upon which as many output-level discriminators are applied. Following the technique proposed in [14], other works adopt the output space adversarial adaptation in combination with additional modules. For example, Chen et al. [15] combine semantic segmentation and depth estimation to boost the adaptation performance. In particular, they provide the domain discriminator with segmentation and depth prediction maps jointly, in order to fully exploit the strong correlation between the two visual tasks. Moreover, Luo et al. [16] enhances the adversarial scheme by a co-training strategy that highlights regions of the input image with high prediction confidence. In this way, the adversarial loss can be effectively tuned by balancing the contribution of each spatial unit, so that more focus is directed towards less adapted areas.

3 Methodology

3.1 Overview of the Model

The model we applied in the project is proposed in [14], as illustrated in Figure 1. This model mainly consists of two part: one is the segmentation network $\mathbf{G}$, the other is the domain adaptation module with discriminators $\mathbf{D_i}$. Images from source and target domains are sent into the segmentation network to obtain output predictions. Then the predictions are passed into the domain adaptation module, where there are some discriminators that tries to distinguish whether the input is from the source or target domain. The domain adaptation module helps us to transfer the model trained on source domain to target domain. The detailed training procedure is introduced in Section 3.3

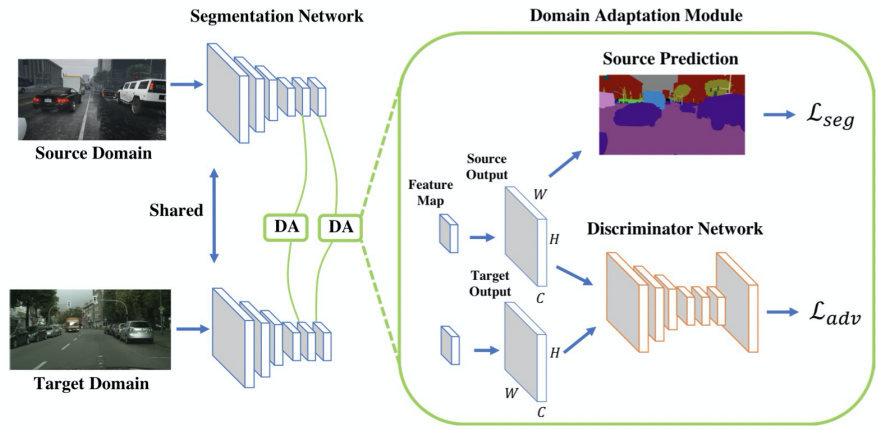

Figure 1: Algorithmic overview.

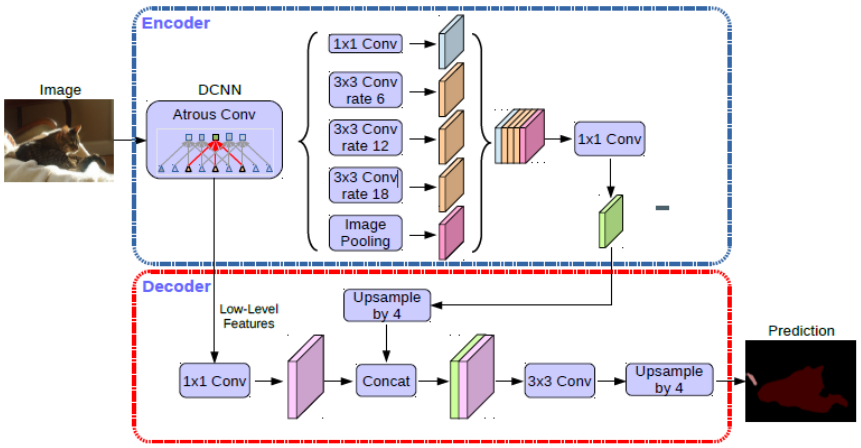

Figure 2: Architecture of DeepLabv3+

3.2 Segmentation Network Architecture

[14] adopts DeepLabv2 framework with ResNet-101 model as its baseline network. In our project, we use the DeepLabv3+ architechture with ResNet-101 model instead and hope to achieve higher quality segmentation results. In the following, we briefly introduce DeepLabv3+. For a comprehensive introduction, one can refer to [6].

According to Figure 2, DeepLabv3+ consists two parts: the encode part and the decode part.

For the encoder part, the network backbone could be ResNet-101 or modifed aligned Xception. The feature maps outputing from the backbone are divided into 2 parts. The lower-level features are sent into decoder directly, while other feature maps are sent into Atrous Spatial Pyramid Pooling (ASPP) part. ASPP applies several parallel atrous convolution with different rates and outputs five feature maps, which are also sent into decoder part after a point-wise convolution.

For the decoder part, after a point-wise convolution or umsamping, the feature maps come from backbone directly and ASPP are concated together. Then after 3×3 convolution and unsampling, DeepLabv3+ finally gets the segmentation.

3.3 Training Procedure

For source predictions, a cross-entropy loss is computed based on the ground truth of source samples as

$$\mathcal{L}_{CE}\left(p_s, y_s\right) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} y_s^{n,c}\log\left(p_s^{n,c}\right)$$

where y_s is the ground truth label of images from source domain. $p_s = \mathbf{G}(x_s)$ is segmentation output of source samples. C is the number of categories.

Then the target predictions, together with the source predictions are sent into the discriminator $\mathbf{D}$ (we suppose the single-level adversary training first). $\mathbf{D}$ tries to distinguish whether the input is from the source or target domain and outputs 0 if the sample is drawn from the target domain, otherwise 1. A cross-entropy loss $\mathcal{L}_d$ is calculated for the discriminator training as

$$\mathcal{L}_d(P) = -\sum_{h,w}(1-z)\log\left(\mathbf{D}(P)^{(h,w,0)}\right) + z\log\left(\mathbf{D}(P)^{(h,w,1)}\right)$$

An adversary loss is calculated as:

$$\mathcal{L}_T\left(x_t\right) = -\sum_{h,w}\log\left(\mathbf{D}\left(P_t\right)^{(h,w,1)}\right)$$

Where x_t and $p_t = \mathbf{G}(x_t)$ represent images from target domain and segmentation output of the target samples respectively. $\mathbf{D}(P_t)$ is the output of the discriminator. This loss is designed to train the segmentation network and fool the discriminator by maximizing the probability of the target prediction being considered as the source prediction.

The adversary loss and the cross-entropy loss of the source samples are used to train the segmentation network. The overall training objective function of the segmentation network can be written as

$$\mathcal{L}\left(x_s, x_t\right) = \lambda_{seg}\mathcal{L}_{CE}\left(p_s, y_s\right) + \lambda_{adv}\mathcal{L}_T\left(x_t\right)$$

For the multilevel adversary training, the training objective function is:

$$\mathcal{L}\left(x_s, x_t\right) = \sum_i \lambda_{seg}^i\mathcal{L}_{CE}^i\left(p_s, y_s\right) + \sum_i \lambda_{adv}^i\mathcal{L}_T^i\left(x_t\right)$$

where i indicates the level used for predicting the segmentation output. For this project, we choose $i = 2$ as in [14].

4 Experiments

4.1 Datasets

Cityscapes is a large-scale database with a focus on semantic understanding of urban street scenes. It contains a diverse set of stereo video sequences recorded in street scenes from 50 cities, with high quality pixel-level annotated frames. It includes semantic and dense pixel annotations of 30 classes, grouped into 8 categories—flat surfaces, humans, vehicles, constructions, objects, nature, sky, and void.

SYNTHIA is a large dataset of photo-realistic frames rendered from a virtual city with precise pixel-level semantic annotations. We use **SYNTHIA-RAND-CITYSCAPES** subset that contains 9400 images with annotations that are compatible with cityscapes.

In task one (segmentaiton model test), we use SYNTHIA and Cityscapes datasets to validate the performance of our segmentation models.

In task two (unsupervised domain adaptation on segmentation), due to the high labor cost of annotating segmentation ground truth, we first adapt the model trained on large-scale synthetic data to real-world data, i.e., Cityscapes. Then we split Cityscapes dataset into two parts, one as source domain and the other as target domain. We perform domain transfer between these two parts and evaluate results. Road scenes in different cities vary, so it's reasonable and convenient to do this way.

4.2 Metrics

Below we list the metrics used for assessing the accuracy of segmentation algorithms. Although quantitative metrics are used to compare different models on benchmarks, the visual quality of model outputs is also important in deciding which model is best (as human is the final consumer of many of the models developed for computer vision applications).

Pixel accuracy simply finds the ratio of pixels properly classified, divided by the total number of pixels. For $K+1$ classes (K foreground classes and the background) pixel accuracy is defined as follows:

$$\text{PA} = \frac{\sum_{i=0}^{K} p_{ii}}{\sum_{i=0}^{K} \sum_{i=0}^{K} p_{ij}}$$

Mean Pixel Accuracy (MPA) is the extended version of PA, in which the ratio of correct pixels is computed in a per-class manner and then averaged over the total number of classes, as in the following:

$$\text{MAP} = \frac{1}{K+1} \sum_{i=0}^{K} \frac{p_{ij}}{\sum_{j=0}^{K} p_{ij}}$$

Intersection over Union (IoU) or the **Jaccard Index** is one of the most commonly used metrics in semantic segmen- tation. It is defined as the area of intersection between the predicted segmentation map and the ground truth, divided by the area of union between the predicted segmentation map and the ground truth:

$$\text{IoU} = J(A, B) = \frac{|A \bigcap B|}{|A \bigcup B|}$$

Mean-IoU is another popular metric, which is defined as the average IoU over all classes. It is widely used in reporting the performance of modern segmentation algorithms.

Precision/Recall/F1 score are popular metrics for reporting the accuracy of many of the classical image segmentation models. Precision and recall can be defined for each class, as well as at the aggregate level, as follows:

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

where TP refers to the true positive fraction, FP refers to the false positive fraction, and FN refers to the false negative fraction. Usually we are interested into a combined version of precision and recall rates. A popular such a metric is called the F1 score, which is defined as the harmonic mean of precision and recall:

$$\text{F1-score} = \frac{2Precision * Recall}{Precision + Recall}$$

4.3 Experiment Setup

Discriminator. The network consists of 5 convolution layers with kernel 4 x 4 and stride of 2, where the channel number is 64, 128, 256, 512, 1, respectively. Except for the last layer, each convolution layer is followed by a leaky ReLU parameterized by 0.2. We do not use any batch-normalization layers as we jointly train the discriminator with the segmentation network using a small batch size.

Segmentation Network. It is essential to build upon a good baseline model to achieve high-quality segmentation results. We adopt the DeepLab-v3+ framework with ResNet-101 model and DeepLab-v2 framework with ResNet-101 model pre-trained on ImageNet as our segmentation baseline networks, which is the same as other works [6, 14].

Before the adaptation, we pre-train the network on the source domain for $70k$ steps to get a high-quality source trained network. We implement the algorithms using PyTorch [17] on a single NVIDIA 1080Ti GPU. We adopt multi-level adaptation model same as [14], i.e., we extract feature maps from the *conv4* layer and add an ASPP module as the auxiliary classifier. Figure 1 shows the multi-level adaptation model in [14]. We use two levels same as [14]. Accordingly, we have two discriminators.

Network Training. Following [14], we train the segmentation network with Stochastic Gradient Descent (SGD) optimizer with learning rate 2.5×10^{-4}, momentum 0.9 and weight decay 5×10^{-4}. We schedule the learning rate using "poly" policy: the learning rate is multiplied by $(1 - \frac{iter}{max\_iter})^{0.9}$ [4]. We employ the random mirror and gaussian blur to augment data, the same as [18]. For training the discriminator, we use the Adam optimizer with the learning rate as 10^{-4} and the same polynomial decay as the segmentation network. The momentum is set as 0.9 and 0.99.

5 Experimental Results

In this section, we present experimental results to validate the domain adaptation method for semantic segmentation under different settings. First, we show evaluations of the model trained synthetic dataset (i.e., SYNTHIA [19]) and test the adapted model on real-world images from the Cityscapes [17] dataset. Several experiments including ablation study are also conducted, e.g., adaptation in the feature/output spaces and single/multi level adversarial learning. Second, we carry out experiment on the dichotomous Cityscapes dataset, where the model is trained on one group of cities and adapted to another group of cities without annotations. In all the experiments, we show the aforementioned metrics and the IoU metric is used for comparison.

5.1 SYNTHIA to Cityscapes

Following the evaluation protocol of other works [20, 21], we evaluate the IoU and mIoU of the shared 16 classes between two datasets and 13 classes excluding the class with *. Table 1 shows the results.

Table 1: Results for SYNTHIA-to-Cityscapes experiments

SYNTHIA→Cityscapes																			
Method	Seg Model	road	sidewalk	building	wall*	fence*	pole*	light	sign	veg.	sky	person	rider	car	bus	motor	bike	mIoU (%)	mIoU* (%)
Source only	DeeplabV2	43.8	19.4	75.8	6.4	0.2	**26.3**	8.6	11.9	**79.0**	**81.1**	**57.6**	**21.4**	50.6	14.0	**20.8**	28.2	34.1	39.4
Source only	DeeplabV3+	55.3	21.4	74.8	5.3	0.1	24.1	12.7	13.5	74.0	71.9	52.0	16.9	53.7	23.8	10.0	22.5	33.2	38.6
AdaptSegNet	DeeplabV2	53.1	**24.0**	64.9	4.7	0.2	25.9	13.6	10.6	76.9	76.3	52.8	18.3	63.6	**31.2**	16.0	**34.7**	35.3	41.1
AdaptSegNet	DeeplabV3+	**62.2**	23.4	**76.8**	**10.1**	**0.3**	25.7	**14.9**	**15.5**	75.5	73.4	53.1	18.6	**66.2**	27.4	19.1	29.9	**37.0**	**42.8**

5.1.1 Overall Results

Table 1 summarizes the experimental results for SYNTHIA-to-Cityscpaes adaptation with different segmentation models. As Table 1 shows, AdaptSegNet with DeeplabV3+ segmentation model achieves the state-of-the-art performace.

On the other hand, we argue that utilizing a stronger segmentation model is critical for understanding the importance of different adaptation components as well as for enhancing the performance to enable real-world applications.

In addition, another factor to evaluate the adaptation performance is to measure how much gap is narrowed between the adaptation model and the fully-supervised model. Hence, we train the model using annotated ground truths in the Cityscapes dataset as the oracle results. Table 2 shows the gap under different segmentation models. We observe that, although the oracle result does not differ a

Table 2: Performance gap between the adapted model and the fully-supervised (oracle) model.

SYNTHIA → Cityscapes				
Method	Seg Model	Adapt	Oracle	mIoU Gap
AdaptSegNet(single)	DeeplabV2	35.3	70.8	-35.5
AdaptSegNet(single)	DeeplabV3+	37.0	71.8	-34.8
AdaptSegNet(multi)	DeeplabV2	36.0	71.4	-35.4

Table 3: Sensitivity analysis of λ_{adv} for output space domain adaptation in the proposed method.

	SYNTHIA → Cityscapes			
λ_{adv}	0.0005	0.001	0.002	0.004
Output	37.0	38.3	38.2	38.3

lot between DeeplabV2 and DeeplabV3+ based models, the gap is larger for the DeeplabV2 one. It suggests that to narrow the gap, using a better model to conduct adversarial domain adaptation is more practical.

5.2 Parameter Analysis

We perform the following investigative experiments on SYNTHIA to Cityscpaes.

Parameter Sensitivity Analysis. We show the sensitivity analysis of parameters λ_{seg} and λ_{adv}. We first consider single-level case. Table shows that a smaller λ_{adv} may not facilitate the training process significantly while a larger λ_{adv} may propagate incorrect gradients to the network. We empirically choose λ_{adv} as 0.001 in the single-level setting.

Single-level v.s. Multi-level Adversarial Learning. We present results of using multi-level and single-level adversarial learning in Table 2. Here, we utilize an additional adversarial module (see Figure 1), we use the same weight as in the single-level setting for the high-level output space (i.e., $\lambda^1_{seg} = 1$ and $\lambda^1_{adv} = 0.001$). Since the low-level output carries less information to predict the segmentation, we use smaller weights for both the segmentation and adversarial loss (i.e., $\lambda^2_{seg} = 0.1$ and $\lambda^2_{adv} = 0.0002$). Evaluation results show that our multi-level adversarial adaptation further improves the segmentation accuracy.

5.3 Between Cityscapes Adaptation

We use one group of cities (aachen, bremen, darmstadt, erfurt, hanover, krefeld, strasbourg, tubingen, weimar) as source domain, and another group of cities (bochum, cologne, dusseldorf, hamburg, jena, monchengladbach, stuttgart, ulm, zurich) as target domain. The source domain contains 1647 images, and the target domain contains 1362 images. We extract 90 images from target domain for evaluation, each city 10 images.

Since a smaller domain gap results in smaller output differences, we use smaller weights for the adversarial loss (i.e. $\lambda^i_{adv} = 0.0005$) when training our models, while the weights for segmentation remain the same as previous experiments.

We show the result in Table 5. Again, our final multi-level model achieves consistent improvement for different cities, which demonstrates the advantages of the proposed adaptation method in the output space.

6 Conclusion

In this report, we tackle the domain adaptation problem for semantic segmentation via adversarial learning in the output space. We adopt the AdaptSegNet method [14] and use two different segmentation models(DeeplabV2 and DeeplabV3+) to compare various results. Experimental results show that DeeplabV3+ model performs favorably against DeeplabV2(which is used in the original paper). Due

Table 4: Sensitivity analysis of λ_{seg} for outupu space domain adaptation in the proposed method.

SYNTHIA → Cityscapes					
λ_{seg}	0.5	0.2	0.1	0.05	0.02
Output	37.0	38.3	38.2	32.8	43.0

Table 5: Results for group0-to-group1 experiments

Group0→Group1																				
Method	Seg Model	road	sidewalk	building	wall	fence	pole	light	sign	veg	terrain	sky	person	rider	car	truck	train	bus	motor	bike	mIoU (%)
Source only	DeeplabV2	95.8	70.9	87.6	40.5	46.2	**46.0**	**48.8**	**51.6**	87.4	55.0	90.5	**67.3**	**44.9**	90.7	45.2	**79.0**	**36.5**	76.7	**56.0**	64.0
Source only	DeeplabV3+	**96.3**	**73.6**	**87.8**	41.1	**47.1**	45.1	46.5	50.0	**87.8**	56.3	**92.6**	63.8	41.7	**91.1**	**81.2**	72.4	33.4	71.1	47.0	**65.0**
AdaptSegNet	DeeplabV2	94.9	66.2	86.0	**42.0**	42.2	39.5	41.2	43.8	85.7	53.2	88.8	65.1	35.0	88.7	37.6	58.8	29.4	**79.5**	53.7	59.5
AdaptSegNet	DeeplabV3+	95.5	70.7	85.8	35.8	42.8	39.6	37.3	40.9	86.4	**57.4**	90.1	61.7	33.0	89.3	62.2	71.1	24.9	68.8	51.3	60.2

to time constraints, we didn't tune the model to its best performance and improve the visual results. This may be left as our future work.

Acknowledgements

Much of this project is based on [14, 22]. We thank Prof. Chi for insightful dicussions and ideas. We also thank North Carolina State University's GEARS program for providing us with such fulfilling research experience.

References

[1] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[2] A. G. Schwing and R. Urtasun, "Fully connected deep structured networks," *arXiv preprint arXiv:1503.02351*, 2015.

[3] 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.

[4] Chen, Liang-Chieh and Papandreou, George and Kokkinos, Iasonas and Murphy, Kevin and Yuille, Alan L, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

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

[6] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801–818.

[7] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International conference on machine learning*. PMLR, 2015, pp. 1180–1189.

[8] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The journal of machine learning research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[9] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7167–7176.

[10] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, "Unsupervised pixel-level domain adaptation with generative adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3722–3731.

[11] 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.

[12] D. Pathak, P. Krahenbuhl, and T. Darrell, "Constrained convolutional neural networks for weakly supervised segmentation," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1796–1804.

[13] Y.-H. Chen, W.-Y. Chen, Y.-T. Chen, B.-C. Tsai, Y.-C. Frank Wang, and M. Sun, "No more discrimination: Cross city adaptation of road scene segmenters," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1992–2001.

[14] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7472–7481.

[15] Y. Chen, W. Li, X. Chen, and L. V. Gool, "Learning semantic segmentation from synthetic data: A geometrically guided input-output adaptation approach," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1841–1850.

[16] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2507–2516.

[17] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[18] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2881–2890.

[19] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3234–3243.

[20] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez, "Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2517–2526.

[21] Y. Zou, Z. Yu, B. Kumar, and J. Wang, "Unsupervised domain adaptation for semantic segmentation via class-balanced self-training," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 289–305.

[22] M. Chen, H. Xue, and D. Cai, "Domain adaptation for semantic segmentation with maximum squares loss," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2090–2099.