Alleviating Semantic-level Shift: A Semi-supervised Domain Adaptation Method for Semantic Segmentation

Zhonghao Wang[1], Yunchao Wei[3], Rogerio Feris[2], Jinjun Xiong[2],
Wen-mei Hwu[1], Thomas S. Huang[1], Honghui Shi[4,1]

[1]C3SR, UIUC, [2]IBM Research, [3]ReLER, UTS, [4]University of Oregon

Abstract

Utilizing synthetic data for semantic segmentation can significantly relieve human efforts in labelling pixel-level masks. A key challenge of this task is how to alleviate the data distribution discrepancy between the source and target domains, i.e. *reducing domain shift. The common approach to this problem is to minimize the discrepancy between feature distributions from different domains through adversarial training. However, directly aligning the feature distribution globally cannot guarantee consistency from a local view (*i.e. *semantic-level). To tackle this issue, we propose a semi-supervised approach named Alleviating Semantic-level Shift (ASS), which can promote the distribution consistency from both global and local views. We apply our ASS to two domain adaptation tasks, from GTA5 to Cityscapes and from Synthia to Cityscapes. Extensive experiments demonstrate that: (1) ASS can significantly outperform the current unsupervised state-of-the-arts by employing a small number of annotated samples from the target domain; (2) ASS can beat the oracle model trained on the whole target dataset by over 3 points by augmenting the synthetic source data with annotated samples from the target domain without suffering from the prevalent problem of overfitting to the source domain.*

1. Introduction

Due to the development and use of deep learning techniques, major progress has been made in semantic segmentation, one of the most crucial computer vision tasks [2, 3, 29, 4, 6, 14, 13]. However, the current advanced algorithms are often data hungry and require a large amount of pixel-level masks to learn reliable segmentation models. Therefore, one problem arises – *annotating pixel-level masks is costly in terms of both time and money.* For example, Cityscapes [7], a real footage dataset, requires over 7,500 hours of human labor on annotating the semantic segmentation ground truth.

To tackle this issue, unsupervised training methods [5, 21, 22, 28, 24] were proposed to alleviate the burdensome annotating work. Specifically, images labeled from other similar datasets (source domain) can be utilized to train a model and adapted to the target domain by addressing the domain shift issue. For the semantic segmentation task on Cityscapes dataset specifically, previous works [19, 20] have created synthetic datasets which cost little human effort to serve as the source datasets.

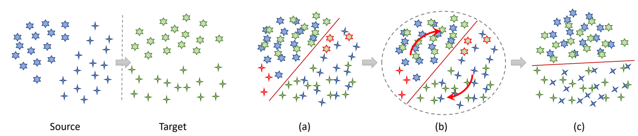

Figure 1: Domain adaptation. (a) Global adaptation. (b) Semantic-level adaptation. (c) Ideal result.

While evaluating the previous unsupervised or weakly-supervised methods for semantic segmentation [22, 27, 26, 12, 11, 25, 18], we found that there is still a large performance gap between these solutions and their fully-supervised counterparts. By delving into the unsupervised methods, we observe that the semantic-level features are weakly supervised in the adaptation process and the adversarial learning is only applied on the global feature representations. However, simply aligning the features distribution from global view cannot guarantee consistency in local view, as show in Figure 1 (a), which leads to poor segmentation performance on the target domain. To address this problem, we propose a semi-supervised learning framework – Alleviating Semantic-level Shift (ASS) model – for better promoting the distribution consistency of features from two domains. In particular, ASS not only adapts global features between two domains but also leverages a few labeled images from the target domain to supervise the segmentation task and the semantic-level feature adaptation task. In this way, the model can ease the inter-class confusion problem during the adaptation process (as shown in Figure 1 (b)) and ultimately alleviate the domain shift from local view (as shown in Figure 1 (c)). As a result, our method 1) is much better than the current state-of-the-art unsupervised methods by using a very small amount

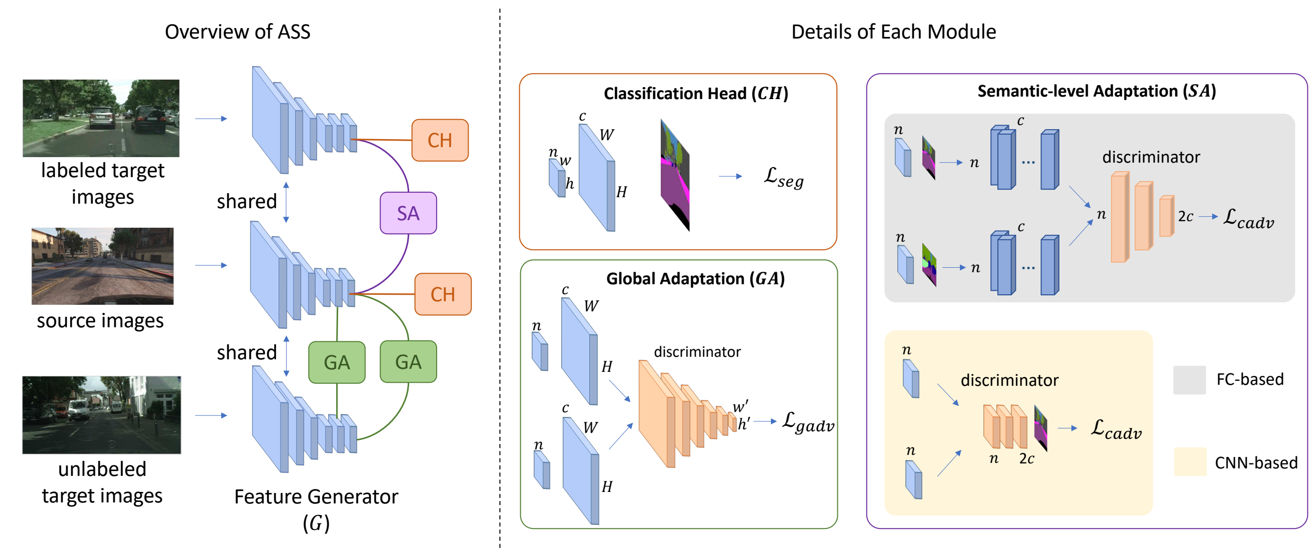

Figure 2: Structure overview. c is the number of classes for adaptation. W and H are the width and height of the input image respectively. n is the number of feature channels of the feature map.

of the labeled target domain images; 2) addresses the prevalent problem that semi-supervised models typically overfit to the source domain [23], and outperforms the oracle model trained with the whole target domain dataset by utilizing the synthetic source dataset and labeled images from the target domain.

2. Related Works

Semantic segmentation. This task requires segment the pixels of images into semantic classes. Deeplab [2, 3, 4] is such a series of deep learning models that attained top on the 2017 Pascal VOC [8] semantic segmentation challenge. It uses Atrous Spatial Pyramid Pooling (ASPP) module which combines multi-rate atrous convolutions and the global pooling technique to enlarge the field of view on the feature map and therefore deepen the model's understanding of the global semantic context. Deeplab v2 has laconic structure and good performance in extracting images features and can be easily trained. Therefore, it is selected as the backbone network for our work.

Domain adaptation. This task requires transfer and apply the useful knowledge of the model trained on the off-the-shelf dataset to the target task dataset [9]. A typical structure for the domain adaptation is Generative Adversarial Networks (GAN) [10]. It consists of a discriminator that distinguishes which domain the input feature maps are from, and a generator that generates the feature maps to fool the discriminator. The discriminator thereby supervises the generator to minimize the discrepancy of the feature representations from the two domains.

3. Method: Alleviating Semantic-level Shift

We randomly select a subset of images from the target domain with ground truth annotations, and denote this set of images as $\{\mathcal{I}_{\mathcal{T}_\mathcal{L}}\}$. We denote the whole set of source images and the set of unlabeled target images as $\{\mathcal{I}_\mathcal{S}\}$ and $\{\mathcal{I}_{\mathcal{T}_\mathcal{U}}\}$ respectively. As shown in Figure 2, our domain adaptation structure has four modules: the feature generation module G, the segmentation classification module CH, the global feature adaptation module GA and the semantic-level adaptation module SA. We denote the output feature maps of G by F, the ground truth label maps by Y and the downsampled label maps (of the same height and width as F) as y. We use H, W to denote the height and width of the input image, h, w to denote those of F, and h', w' to denote those of the confidence maps output by the discriminator of GA. C is the class set, c is the number of classes, and n is the channel number of F. When testing the model, we forward the input image to G and use CH to operate on F to predict the semantic class that each pixel belongs to. The following sections will introduce the details of each module.

3.1. Segmentation

We forward F to a convolutional layer to output the score maps with c channels. Then, we use a bilinear interpolation to upsample the score maps to the original input image size and apply a softmax operation channel-wisely to get score maps P. The segmentation loss L_{seg} is calculated as

$$L_{seg}(I) = -\sum_{H,W}\sum_{k\in C} Y^{(H,W,k)}\log(P^{(H,W,k)}) \quad (1)$$

3.2. Global Feature Adaptation Module

This module adapts F from the source domain to the target domain. we input the source image score maps P_s to the discriminator D_g of GA to conduct the adversarial training. We define the adversarial loss as:

$$L_{gadv}(I_s) = -\sum_{h',w'} \log(D_g(P_s)^{(h',w',1)}) \tag{2}$$

We define 0 as the source domain pixel and 1 as the target domain pixel for the output of D_g. Therefore, this loss will force G to generate features closer to the target domain globally. To train D_g, we forward P_s and P_{t_u} to D_g in sequence. The loss of D_g is calculated as:

$$\begin{aligned} L_{gd}(P) = -\sum_{h',w'} ((1-z)\log(D_g(P_s)^{(h',w',0)}) \\ +z\log(D_g(P_t)^{(h',w',1)})) \end{aligned} \tag{3}$$

where $z = 0$ if the feature maps are from the source domain and $z = 1$ if the feature maps are from the target domain.

3.3. Semantic-level Adaptation Module

This module adapts the feature representation for each class in the source domain to the corresponding class feature representation in the target domain to alleviate the domain shift from semantic-level.

3.3.1 Fully connected semantic adaptation (FCSA)

We believe that the feature representation for a specific class at each pixel should be close to each other. Thereby, we can average these feature vectors across the height and width to represent the semantic-level feature distribution, and adapt the averaged feature vectors to minimize the distribution discrepancy between two domains. The semantic-level feature vector V_k of class k is calculated as

$$V^k = \frac{\sum_{h,w} y^{(h,w,k)} F^{(h,w,:)}}{\sum_{h,w} y^{(h,w,k)}} \tag{4}$$

where $k \in C$, $V^k \in \mathbb{R}^n$. Then we forward these semantic-level feature vectors to the semantic-level feature discriminator D_s for adaptation, as shown in Figure 2. D_s only has 2 fully connected layers, and outputs a vector of $2c$ channels after a softmax operation. The first half and the last half channels correspond to classes from the source domain and the target domain respectively. Therefore, the adversarial loss can be calculated as

$$L_{sadv}(I_s) = -\sum_{k \in C} \log(D_s(V_s^k)^{(k+c)}) \tag{5}$$

To train D_s, we let it classify the semantic-level feature vector to the correct class and domain. The loss of D_s can be calculated as:

$$\begin{aligned} L_{sd}(V) = -\sum_{k \in C} ((1-z)\log(D_s(V^k)^{(k)}) \\ +z\log(D_s(V^k)^{(k+c)})) \end{aligned} \tag{6}$$

where $z = 0$ if the feature vector is from the source domain and $z = 1$ if it is from the target domain.

3.3.2 CNN semantic adaptation (CSA)

We observe that it is hard to extract the semantic-level feature vectors, because we have to use the label maps to filter pixel locations and generate the vectors in sequence. Therefore, inspired from the previous design, we come up with a laconic CNN semantic-level feature adaptation module. The discriminator uses convolution layers with kernel size 1×1, which acts as using the fully connected discriminator to operate on each pixel of F, as shown in Figure 2. The output has $2c$ channels after a softmax operation where the first half and the last half correspond to the source domain and the target domain respectively. Then, the adversarial loss can be calculated as:

$$L_{sadv}(I_s) = -\sum_{h,w} \log(D_s(F_s)^{(h,w,k+c)}) \tag{7}$$

where k is the pixel ground truth class. To train the discriminator, we can use the loss as follows:

$$\begin{aligned} L_{sd}(F) = -\sum_{h,w}((1-z)\log(D_s(F)^{(h,w,k)}) \\ +z\log(D_s(F)^{(h,w,k+c)})) \end{aligned} \tag{8}$$

3.4. Adversarial Learning Procedure

Our ultimate goal for G is to have a good semantic segmentation ability by adapting features from the source domain to the target domain. Therefore, the training objective for G can derive from Eqn (1) as

$$\begin{aligned} L(I_s, I_{t_l}) = \lambda_{seg}(L_{seg}(I_s) + L_{seg}(I_{t_l})) \\ +\lambda_{gadv} L_{gadv}(I_s) + \lambda_{sadv} L_{sadv}(I_s) \end{aligned} \tag{9}$$

where λ is the weight parameter. The two discriminators should be able to distinguish which domain the feature maps are from, which enables the features to be adapted in the right direction. We can simply sum up the two discriminator losses as the training objective for discriminative modules.

$$\begin{aligned} L(F_s, F_{t_u}, F_{t_l}) = \lambda_{gd}(L_{gd}(F_s) + L_{gd}(F_{t_u})) \\ +\lambda_{sd}(L_{sd}(F_s) + L_{sd}(F_{t_l})) \end{aligned} \tag{10}$$

In summary, we will optimize the following min-max criterion to let our model perform better in segmentation task by adapting the features extracted from the source domain more alike the ones extracted from the target domain.

$$\max_{D_g, D_s} \min_{G} L(I_s, I_{t_l}) - L(F_s, F_{t_u}, F_{t_l}) \tag{11}$$

4. Implementation

4.1. Network Architecture

We follow [22] to build the network structures for the backbone network, the classification module (CH) and the

Table 1: GTA5 $\rightarrow$ Cityscapes: performance contributions of adaptation modules. The oracle model is only trained with the given number of Cityscapes labeled images.

# City	Oracle	GA	GA+FCSA	GA+CSA	Improve
0	-	42.4	-	-	-
50	39.5	50.0	50.2	50.1	+10.6
100	43.6	53.5	54.1	54.2	+10.6
200	47.1	54.4	56.4	56.0	+8.9
500	53.6	56.5	59.9	60.2	+6.6
1000	58.6	58.0	63.8	64.5	+5.9
2975 (all)	65.9	59.71	68.8	69.1	+3.2

Table 2: parameters analysis

# City	$\lambda = 1$	$\lambda = 0.2$	$\lambda = 0.04$	$\lambda = 0.008$
100	**54.11**	53.87	53.68	53.96
500	59.76	59.29	**59.89**	59.74

(a): λ_{sadv} for fully connected semantic-level adaptation module

# City	$\lambda = 1$	$\lambda = 0.1$	$\lambda = 0.01$	$\lambda = 0.001$
500	59.76	59.46	**60.16**	59.67

(b): λ_{sadv} for CNN semantic-level adaptation module

Table 3: SYNTHIA $\rightarrow$ Cityscapes: performance contributions of adaptation modules.

# City	Oracle	GA	GA+CSA	Improve
0	-	46.7	-	-
50	52.6	60.7	57.4	+8.1
100	57.6	62.1	58.3	+4.5
200	60.8	64.8	64.5	+4.0
500	66.5	69.1	69.8	+3.3
1000	70.7	71.8	73.0	+2.3
2975 (all)	73.8	75.0	77.1	+3.3

global adaptation module (GA). For $FCSA$, we use two fully connected layers with channel number of 1024 and put a Leaky ReLU [16] of 0.2 negative slope between them, and twice the class number for the output. For CSA, we use two convolutional layers with the kernel size of 1×1, stride of 1 and channel number of 1024 and twice the class number for the output. We insert a Leaky ReLU [16] layer with 0.2 negative slope between the two convolutional layers.

4.2. Network Training

We optimize Eqn (11) in an adversarial strategy. We first . We use Stochastic Gradient Descent (SGD) with Nesterov's method [1] with momentum 0.9 and weight decay 5×10^{-4} to optimize the segmentation network . Following [2], we set the initial learning rate to be 2.5×10^{-4} and let it polynomially decay with the power of 0.9. We use Adam optimizer [15] with momentum 0.9 and 0.99 for all the discriminator networks. We set the initial learning rate to be 10^{-4} and follow the same polynomial decay rule.

5. Experiments

We validate the effectiveness of our proposed method by transferring our model from a synthetic dataset (GTA5 [19] or SYNTHIA [20]) to a real-world image dataset Cityscapes [7]. The Cityscapes dataset contains 2975 images for training and 500 images for validation with 19-class fine-grained semantic annotations. following [22], we first trained our model on the GTA5 dataset containing 19466 images and Cityscapes training set images and tested on the Cityscapes validation set for the whole 19 classes. The result is shown in Table 1. First, notice that the current state-of-the-art unsupervised model achieves 48.5 in mIoU [17]. Our model can beat it by adding 50 Cityscapes images into the training process. This proves our argument that the model can have significant improvement by adding a few target domain information. Second, the contribution of GA module disappears or is negative when the labeled Cityscapes images reach a number of 1000 or more compared to the oracle models. This is because the model with weak adaptation supervision overfits to the source domain so that it does not help much by adding relatively few more target images for the training process. However, the models $GA + FCSA$ and $GA + CSA$ both have on-par improvements if trained with over 50 Cityscapes labeled images. We argue that this is due to the strong adaptation supervision. Shown in Table 2, we observe that the CSA and $FCSA$ structures are not very sensitive to the hyperparameters. We also provide some visualization results in Figure 3. Because CSA is more laconic than $FCSA$, we only compare the model $GA+CSA$ with the other baseline models on transferring from Synthia dataset containing 9400 images to Cityscapes dataset. We compare the mIoU of 13 classes shared between SYNTHIA and Cityscapes [22] as shown in Table 3. The results can further support our arguments above.

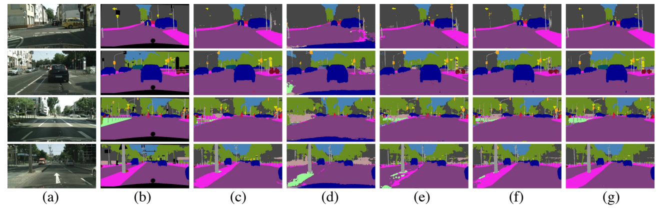

Figure 3: (a) image; (b) ground truth; (c) oracle model trained with the whole Cityscapes dataset; (d) unsupervised; (e) ours+200city; (f) ours+1000city; (g) ours+wholecity

6. Conclusion

This paper proposes a semi-supervised learning framework to adapt the global feature and the semantic-level feature from the source domain to the target domain for the semantic segmentation task. As a result, with a few labeled target images, our model outperforms current state-of-the-art unsupervised models by a great margin. Our model can also beat the oracle model trained on the whole dataset from target domain by utilizing the synthetic data with the whole target domain labeled images without suffering from the prevalent problem of overfitting to the source domain.

Acknowledgment This work is supported by IBM-UIUC Center for Cognitive Computing Systems Research(C3SR).

References

[1] Aleksandar Botev, Guy Lever, and David Barber. Nesterov's accelerated gradient and momentum as approximations to regularised update descent. In *IEEE IJCNN*, pages 1899–1903, 2017.

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

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE TPAMI*, 40(4):834–848, 2017.

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

[5] Yuhua Chen, Wen Li, and Luc Van Gool. Road: Reality oriented adaptation for semantic segmentation of urban scenes. In *IEEE CVPR*, 2018.

[6] Bowen Cheng, Liang-Chieh Chen, Yunchao Wei, Yukun Zhu, Zilong Huang, Jinjun Xiong, Thomas S Huang, Wen-Mei Hwu, and Honghui Shi. Spgnet: Semantic prediction guidance for scene parsing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5218–5228, 2019.

[7] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *IEEE CVPR*, 2016.

[8] Mark Everingham, S. M. Ali Eslami, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 111(1):98–136, 2015.

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

[10] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks. *NIPS*, 2014.

[11] Qibin Hou, PengTao Jiang, Yunchao Wei, and Ming-Ming Cheng. Self-erasing network for integral object attention. In *NIPS*, 2018.

[12] Zilong Huang, Xinggang Wang, Jiasi Wang, Wenyu Liu, and Jingdong Wang. Weakly-supervised semantic segmentation network with deep seeded region growing. In *IEEE CVPR*, 2018.

[13] Zilong Huang, Yunchao Wei, Xinggang Wang, Honghui Shi, Wenyu Liu, and Thomas S Huang. Alignseg: Feature-aligned segmentation networks. *arXiv preprint arXiv:2003.00872*, 2020.

[14] Jianbo Jiao, Yunchao Wei, Zequn Jie, Honghui Shi, Rynson WH Lau, and Thomas S Huang. Geometry-aware distillation for indoor semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2869–2878, 2019.

[15] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *ICLR*, 2014.

[16] Andrew L. Maas, Awni Y Hannun, and Andrew Ng. Rectifier nonlinearities improve neural network acoustic models. In *ICML*, 2013.

[17] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[18] Rui Qian, Yunchao Wei, Honghui Shi, Jiachen Li, Jiaying Liu, and Thomas Huang. Weakly supervised scene parsing with point-based distance metric learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8843–8850, 2019.

[19] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *ECCV*, 2016.

[20] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio Lopez. The SYNTHIA Dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *IEEE CVPR*, 2016.

[21] Swami Sankaranarayanan, Yogesh Balaji, Arpit Jain, Ser-Nam Lim, and Rama Chellappa. Unsupervised domain adaptation for semantic segmentation with gans. In *IEEE CVPR*, 2018.

[22] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *IEEE CVPR*, 2018.

[23] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018.

[24] Zhonghao Wang, Mo Yu, Yunchao Wei, Rogerior Feris, Jinjun Xiong, Wen mei Hwu, Thomas S. Huang, and Honghui Shi. Differential treatment for stuff and things: A simple unsupervised domain adaptation method for semantic segmentation. *arXiv preprint arXiv:2003.08040*, 2020.

[25] Yunchao Wei, Jiashi Feng, Xiaodan Liang, Ming-Ming Cheng, Yao Zhao, and Shuicheng Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *IEEE CVPR*, 2017.

[26] Yunchao Wei, Xiaodan Liang, Yunpeng Chen, Xiaohui Shen, Ming-Ming Cheng, Jiashi Feng, Yao Zhao, and Shuicheng Yan. Stc: A simple to complex framework for weakly-supervised semantic segmentation. *IEEE TPAMI*, 39(11):2314–2320, 2017.

[27] Yunchao Wei, Huaxin Xiao, Honghui Shi, Zequn Jie, Jiashi Feng, and Thomas S. Huang. Revisiting dilated convolution: A simple approach for weakly- and semi-supervised semantic segmentation. In *IEEE CVPR*, 2018.

[28] Yiheng Zhang, Zhaofan Qiu, Ting Yao, Dong Liu, and Tao Mei. Fully convolutional adaptation networks for semantic segmentation. In *IEEE CVPR*, 2018.

[29] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *IEEE CVPR*, 2017.