# ADVERSARIAL DOMAIN ADAPTATION WITH A DOMAIN SIMILARITY DISCRIMINATOR FOR SEMANTIC SEGMENTATION OF URBAN AREAS

*Liang Yan*[1,2]*, Bin Fan*[1,2]*, Shiming Xiang*[1,2] *and Chunhong Pan*[1]

[1] National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences
[2] School of Artificial Intelligence Institute, University of Chinese Academy of Sciences
{liang.yan, bfan, smxiang and chpan}@nlpr.ia.ac.cn

## ABSTRACT

Existing semantic segmentation models of urban areas have shown to perform well in a supervised setting. However, collecting lots of annotated images from each city to train such models is time-consuming or difficult. In addition, when transferring the segmentation model from the trained city (source domain) to an unseen city (target domain), the performance will largely degrade due to the domain shift. For this reason, we propose a domain adaptation method with a domain similarity discriminator to eliminate such domain shift in the framework of adversarial learning. Contrary to the single-input adversarial network, our domain similarity discriminator, which consists of a Siamese network, is able to measure the similarity of the pairwise-input data. In this way, we can use more information about the pairwise-input to measure the similarity between different distributions so as to address the problem of domain shift. Experimental results demonstrate that our approach outperforms the competing methods on three different cities.

***Index Terms***— domain adaptation, domain shift, semantic segmentation, Siamese network, urban areas

## 1. INTRODUCTION

Semantic segmentation of urban areas is an very important task in image processing, especially for the very high-resolution remote sensing images. It has great significance in the fields of infrastructure planning, land planning, and urban area change detection. With sufficient annotated training images, existing semantic segmentation models, such as [1, 2, 3, 4, 5], have already demonstrated great performance. However, employing the segmentation models trained only with the labeled images of one city to segment the unlabeled images from other cities has shown to be infeasible or ineffective.

The reasons for this phenomenon, which called domain shift in literature [6], include the different architectural styles across different cities and different spectral bands and Ground Sample Distance (GSD) among different imaging sensors. Since it is unpractical to obtain adequate training samples for all the studied cities, researchers are aiming to study domain adaptation techniques to eliminate domain shift without the need of labeling new image datasets.

Domain adaptation is mainly to learn the invariant representations between different domains so as to eliminate the domain shift. There are many domain adaptation methods have been proposed, such as Transfer Component Analysis (TCA) [7], Joint Distribution Adaptation (JDA) [8], Maximum Mean Discrepancy (MMD) [9, 10, 11] and adversarial adaptation methods (DANN [12] and ADDA [13]). TCA and JDA use traditional methods to extract features, then use margin adaptation or joint adaptation to express the differences between different distributions, and then perform domain adaptation tasks. MMD maps the data distribution of different domains to a reproducing kernel Hilbert space (RKHS) and then measures the distance between two distributions in RKHS to reduce the domain shift. The Generative Adversarial Nets (GANs) [14] is adopted in DANN and ADDA where the generator produces simulative features while the discriminator distinguishes them. Among the recent methods that apply domain adaptation to semantic segmentation, FCNs in the wild [15] is the first framework that extends the idea of DANN to solve semantic segmentation problems. Curriculum domain adaptation [16] proposes curriculum-style learning approach to transfer knowledge across different domains. Saito et al. [17] proposed to utilize task-specific classifiers as discriminators to align distributions of source and target domains.

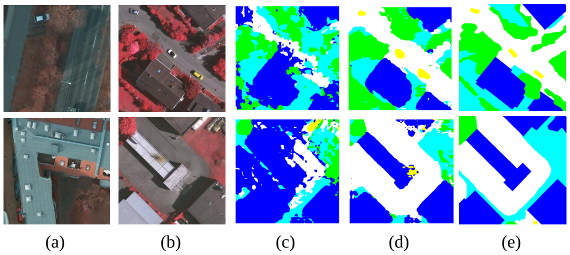

**Fig. 1**: Domain adaptation for semantic segmentation. (a) denotes source images sampled from ISPRS Postdam, (b) denotes target images sampled from ISPRS Vaihingen. (c) denotes the results before using domain adaptation method, (d) denotes the results after using our method, (e) denotes the ground truth of the target images.

Most domain adaptation methods only use the images of the source and target domains separately, and do not consider the similarity between the same domain and the dissimilarity between different domains. By measuring the similarity of two domains, we can utilize that information to regularize the network. However, measuring the similarity between two domains remains an open problem. Bromley et al. [18] used a Siamese network to obtain two different feature maps and then applied a similarity measure function to compute the similarity. SRPN [19] integrates the measurement function into the Siamese network and adapts to minimize the training objective with end-to-end optimization. SD-GAN [20] utilizes a Siamese network as a discriminator to measure the similarity of a pair of images for face verification. The above methods just measure the similarity between the generated images and real images and do not compute the similarity of different domains.

In this paper, we propose an adversarial domain adaptation method with a domain similarity discriminator (DSD), which con-

This work was supported by the National Natural Science Foundation of China under Grants 91646207, 61573352

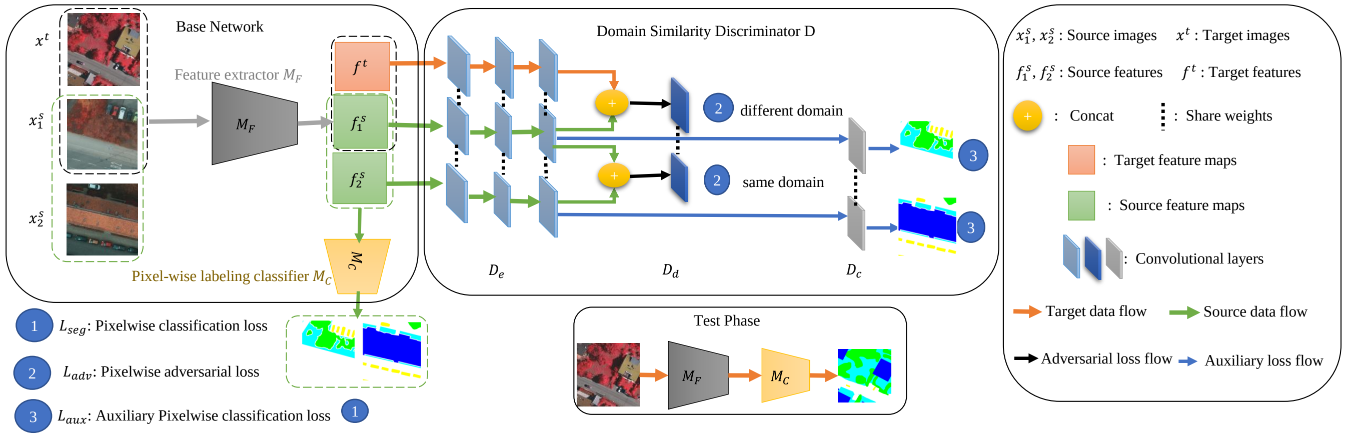

**Fig. 2**: The architecture of the proposed approach for domain adaptation. The architecture consists of two convolution networks. The base network is a semantic segmentation network, such as DeepLabV3, and we split it into two parts as feature extractor $M_F$ and pixel-wise labeling classifier $M_C$. The domain similarity discriminator is composed by four convolution layers, and takes a triplet of feature maps as input to compute their similarities. In the test phase for target data, only $M_F$ and $M_C$ are used. Best viewed in color.

sists of a Siamese network, to eliminate the domain shift for semantic segmentation of urban areas (the model is shown in Fig. 2). Our proposed model utilizes the similar information between the same domain and the dissimilar information between different domains. By integrating DSD and any feature extractor from existing semantic segmentation networks (such as DeepLabV3 [5] used in this paper) into adversarial learning framework, our model can transfer knowledge across different cities. The contributions of this paper are summarized as follows:

- We propose a Siamese-based domain similarity discriminator to distinguish the domain similarity of the pairwise-input feature maps. Therefore, we can simultaneously use the similarity between the same domain and the dissimilarity between different domains. Such information is used to add additional constraints to regularize the network.
- We integrate the domain similarity discriminator and the feature extractor of existing semantic segmentation models into the adversarial learning framework. As a result, the feature extractor can produce domain-invariant features to eliminate domain shift. Experimental results across three different cities show that our approach can achieve better performance than other competing methods.

## 2. ADVERSARIAL DOMAIN ADAPTATION METHOD

In this section, we describe our adversarial domain adaptation method for semantic segmentation of urban areas with the proposed domain similarity discriminator. We consider having a source domain $S$, with both image space $X^s$ and label space $Y^s$. Meanwhile, we have a target domain $T$, with image space $X^t$, but no annotations. We denote $\mathbf{x^s} \in \mathbb{R}^{w \times h \times c}$ and $\mathbf{x^t} \in \mathbb{R}^{w \times h \times c}$ as images sampled from source and target domain. Where $w$ and $h$ are the width and height of the image, and $c$ is the number of channels.

Given a triplet of images $(\mathbf{x^t}, \mathbf{x_1^s}, \mathbf{x_2^s})$ as input, our proposed architecture can be decoupled into three major components. The first part is a feature extractor $M_F(\mathbf{x})$ that transforms each of the input image to a semantic feature space. To simplify the expression, we denote $(\mathbf{f^t}, \mathbf{f_1^s}, \mathbf{f_2^s})$ as the feature maps of $(\mathbf{x^t}, \mathbf{x_1^s}, \mathbf{x_2^s})$, respectively. The second component is a pixel-wise labeling classifier $M_C(M_F(\mathbf{x^s}))$ that classifies the feature maps of source images to label space. The third components, a domain similarity discriminator $D$ performs two different tasks when receiving a pair of feature maps: (1) It distinguishes the similarity of domains between the pairwise-input feature maps. (2) It performs a pixel-wise labeling classification task similar to that of the $M_C$ network. We integrate $D$ and $M_F$ into the adversarial learning framework to make $M_F$ produce domain-invariant features. The first two parts consist of our base network and we treat the third part as a discriminator.

### 2.1. The Base Network

We utilize DeepLabV3 [5] as the base network and split it into two parts: the feature extractor $M_F$ and the pixel-wise classifier $M_C$, as shown in the left of Fig. 2. To ensure that our network performs well on source images, which is known to be effective for the final semantic segmentation task, we should optimize the standard supervised segmentation objective on the source domain. Therefore, we use a pixel-wise softmax cross-entropy loss to achieve this goal:

$$L_{seg} = -\mathbb{E} \sum_{k=1}^{K} \mathbb{1}_{[k=\mathbf{y^s}]} \log M_C(M_F(\mathbf{x^s})), \qquad (1)$$

where $K$ is the number of classes and $\mathbf{x^s}$ and $\mathbf{y^s}$ denote source image and the corresponding label. $\mathbb{1}_{[k=y^s]}$ is an indicator function, which takes 1 when $[k = y^s]$, and 0 otherwise. Note that our method may be generally applied to any semantic segmentation framework.

### 2.2. The Domain Similarity Discriminator

Recently, domain adversarial learning frameworks [15, 17] have been applied for solving domain adaptation in image semantic segmentation problems. Those methods only used the images of the source and target domains separately, and consider neither the similar information between the same domain nor the dissimilar information between different domains. As a comparison, our method effectively utilizes those information.

To achieve this goal,we propose a domain similarity network as a discriminator $D$, which takes a triplet of feature maps $(\mathbf{f^t}, \mathbf{f_1^s}, \mathbf{f_2^s})$ as input. Then this triplet is split into two different two-tuples, defining $(\mathbf{f^t}, \mathbf{f_1^s})$ as a two-tuple from different domains and $(\mathbf{f_1^s}, \mathbf{f_2^s})$ as from the same domain. As shown in Fig. 2, $D$ performs two different tasks when receiving a two-tuple: (1) It distinguishes the received two-tuple whether from the same domain or not. (2) It performs a pixel-wise classification task similar to the $M_C$ network for source feature maps contained in the received two-tuple, which is beneficial to make a stable adversarial training [21].

For the first task, we introduce the Siamese network which is similar to [20], as a pixel-wise domain similarity discriminator $D$. The Siamese network is equivalent to a similarity function that measures the similarity of the distributions of the received two-tuple, as shown in the top right of Fig. 2. We first separately encode each feature maps in the input triplet using the same convolution neural network $D_e$. And then, in order to let $D$ distinguish the received two-tuple whether from the same domain or not, we stack the feature maps of $D_e(\mathbf{f_1^s})$ and $D_e(\mathbf{f_2^s})$ (or $D_e(\mathbf{f^t})$ and $D_e(\mathbf{f_1^s})$) along the channel axis. After that, another convolution layer $D_d$ is applied to aggregate information from the two-tuple, to output a 2-dimensional (2-D) feature map. In this way, $D$ acts as a pixel-wise domain similarity discriminator, which discriminates each pixel in the output 2-D feature map whether zfrom the same domain (with a label of 1) or not (with a label of 0).

To eliminate domain shift, we integrate $D$ and $M_F$ into the adversarial learning framework. Instead of using the minimax loss, the standard loss is adopted to train the generator with inverted labels [14]. In our method, the adversarial loss $L_{adv}$ is split into two independent parts, one for $D$ and the other for $M_F$, which can be described as follows:

$$\begin{aligned} L_{adv,D} = &- \mathbb{E}[\log D_d(D_e(\mathbf{f_1^s}), D_e(\mathbf{f_2^s}))] \\ &- \mathbb{E}[\log(1 - D_d(D_e(\mathbf{f_1^s}), D_e(\mathbf{f^t})))], \end{aligned} \quad (2)$$

$$L_{adv,M_F} = -\mathbb{E}[\log D_d(D_e(\mathbf{f_1^s}), D_e(\mathbf{f^t}))], \quad (3)$$

where $L_{adv,D}$ and $L_{adv,M_F}$ denote pixel-wise adversarial losses for $D$ and $M_F$, respectively. The output of $D_d(\cdot)$ indicates the probability that the elements in the two-tuple are discriminated as belonging to the same domain.

For the second task, we input the feature maps of $D_e(\mathbf{f_1^s})$ and $D_e(\mathbf{f_2^s})$ to another convolution layer $D_c$ and output two predicted label maps. In this way, $D$ acts as a pixel-wise labeling classifier similar to the $M_C$ network. Formally, it can be described as:

$$L_{aux} = -\mathbb{E}\sum_{k=1}^{K} \mathbb{1}_{[k=\mathbf{y^s}]} \log D_c(D_e(\mathbf{f^s})), \quad (4)$$

where $\mathbf{f^s}$ denotes the source images feature maps, such as $\mathbf{f_1^s}$ or $\mathbf{f_2^s}$ and $K$ is the number of classes.

### 2.3. Training steps

To sum up, there are three learning objectives for our method: (1) $M_C$ and $M_F$ should segment source images accurately to ensure that our model does not diverge too far from the source solution and obtain discriminative features. (2) $D$ should distinguish the received two-tuple whether from the same domain or not and perform as a pixel-wise labeling classifier similar to the $M_C$ network for source feature maps. (3) $M_F$ should fool $D$ to distinguish the two-tuple from different domains as coming from the same domain so as to produce domain-invariant features. To achieve these goals, the training within a mini-batch consists of the following three steps.

**Update $M_F$ and $M_C$.** During this step, the parameters of $M_F$ and $M_C$ will be updated while the parameters of $D$ are fixed. The network is trained to minimize $L_{seg}$ as follows:

$$\min_{M_F, M_C} L_{seg}. \quad (5)$$

**Update $D$.** In this step, the parameters of $M_F$ and $M_C$ are fixed while those of $D$ will be updated to distinguish the consistency of domains. To make the training procedure stable, we add an auxiliary pixel-wise loss in this step. The learning objective is as follows:

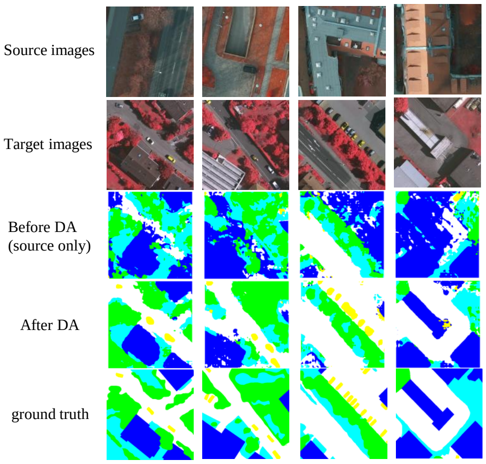

**Fig. 3**: Qualitative results on domain adaptation from POT to VAI.

$$\min_{D} L_{adv,D} + \alpha L_{aux}, \quad (6)$$

where $\alpha$ is a trade-off parameter between $L_{adv,D}$ and $L_{aux}$.

**Update $M_F$.** We train $M_F$ to fool $D$ such that $D$ discriminates the two-tuple from different domains as coming from the same domain so as to make $M_F$ produces domain-invariant features. We achieve this by minimizing $L_{adv,M_F}$, at the same time fixing the parameters of $D$ and $M_C$. We add $L_{aux}$ to stable the training, as well. The learning objective is as follows:

$$\min_{M_F} \beta L_{adv,M_F} + \alpha L_{aux}, \quad (7)$$

where $\beta$ is weight for the adversarial loss of $M_F$.

In the training procedure, the first step is pre-trained for an epoch, and then all the three steps are iteratively trained.

## 3. EXPERIMENTAL RESULTS

In this section, we evaluate our approach across images of three different urban areas which were captured from different locations and with different GSD and spectra.

### 3.1. Datasets

**ISPRS Vaihingen (VAI).** [22] consists of 3-band IRRG (Infrared, Red, Green) image data acquired by airborne sensors, there are 16 annotated images with high resolution about 2500 × 2000 pixels at a GSD of 9cm. We randomly sampled 10 images as training set, and the rest as testing set. We cropped training set to a number of 512 × 512 patches with overlap of 300 pixels (no overlap in testing set). Finally, there are 2152 images in training set and 423 images in testing set.

**ISPRS Postdam (POT).** [22], which consists of 3-band IRRG (Infrared, Red, Green) image data acquired by airborne sensors, there are 24 annotated images with high resolution about 6000 × 6000 pixels at a GSD of 5cm. We randomly sampled 15 images as training set, and the rest as testing set. We cropped training set to a number of 512 × 512 patches with overlap of 200 pixels (no overlap in testing set). In this way, we produced 6000 images for training and 1296 images for testing.

**BeiJing (BEJ).** The BeiJing dataset used in this paper is satellite images collected by ourselves from BaiDu Map with a GSD of 30cm, which consists of 3-band RGB image data. There are 4716 images of 512 × 512 pixels and we randomly sampled 800 images as testing set with the rest as training set.

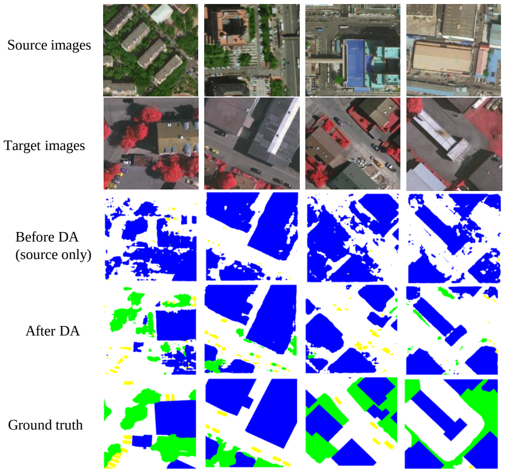

**Fig. 4**: Qualitative results on domain adaptation from BEJ to VAI.

Although six different domain adaptation experiments can be formed by combining any two cities datasets, due to space limits, we chose to show here the two typical and generalized experiments: POT to VAI and BEJ to VAI. We compared the proposed approach with three competing methods: DANN [12], ADDA [13] and MCD-UDA [17]. Since these methods do not release their codes or not be applied to segmentation problem, we implement them by ourselves.

**Implementation details.** All experiments are conducted utilizing the PyTorch [23] framework. For fair comparison, in all experiments we used the DeepLabV3 [5] as our base network, which is pre-trained on ImageNet [24]. Following [5], we adopt the Intersection over Union (IoU) as evaluation criterion: $IoU(P_m, P_{gt}) = \frac{|P_m \bigcap P_{gt}|}{|P_m \bigcup P_{gt}|}$, where $P_{gt}$ denotes ground truth and $P_m$ denotes the prediction map. We trained our network with 50,000 iterations using Adam solver [25] with learning rate $2e-5$ and betas 0.5 and 0.999. We used a batch size of 2 for source domain and 1 for target domain, and the hyper-parameters were set as: $\alpha = 0.1, \beta = 1$.

**Table 1**: The performance of adaptation from POT to VAI(%). road: impervious surface, veg: low vegetation. The symbol $*$ denotes that the code we implement ourselves.

| | car | roof | tree | veg | road | mIoU |
|---|---|---|---|---|---|---|
| Souce-only | 6.0 | 46.6 | 42.0 | 23.9 | 27.5 | 29.2 |
| DANN$^*$ [12] | 33.1 | 68.3 | 55.5 | 26.9 | 64.0 | 49.5 |
| ADDA$^*$ [13] | 30.4 | 67.6 | 43.1 | **29.7** | 62.3 | 46.7 |
| MCD-UDA$^*$ [17] | 8.3 | 52.9 | 32.0 | 25.3 | 55.0 | 34.7 |
| Ours | **38.0** | **70.8** | 53.3 | 29.6 | **65.6** | **51.5** |
| Target-only | 68.2 | 87.0 | 77.2 | 60.0 | 81.1 | 74.7 |

### 3.2. Postdam to Vaihingen

In this experiment, we treated the POT dataset as our source domain, and VAI dataset as our target domain. The domain shift is mainly due to different architectural styles between two cities and different GSD of these datasets. In the training procedure we only used the ground truth of POT dataset, and randomly sampled 400 images from VAI's training set as validation set. We tested our model on the testing set of VAI. POT and VAI contains the same five classes: impervious surface, building roof, low vegetation, tree and car.

Table 1 reports the performance of our method in comparison with [12, 13, 17]. The baseline method is a source-only model, which only uses source domain data for training and tests on target data, achieving a mean IoU (mIoU) of 29.2%. On the contrary, the target-only means training target domain with annotations, which is not the case of domain adaptation but its performance can be considered as a upper bound for the domain adaptation performance. The results show that all the domain adaptation methods achieve better results than the baseline, showing the effectiveness of domain adaptation. Among all these methods, our method performs the best and improves the baseline significantly from 29.2% to 51.5%. The second best results are achieved by DANN with mIoU as 49.5%, 2% lower than ours. In order to prove the validity of our domain similarity discriminator, we remove the auxiliary pixel-wise loss by setting $\alpha = 0$. Experimental result shows that our method achieves a mean IoU of 50.2%, which is still higher than DANN's 49.5%. The qualitative results are shown in Fig. 3. From the segmentation results, we can see that the source-only model seems to suffer from domain shift seriously. Our method can produce better segmentation results, especially for cars, roofs, and roads.

### 3.3. BeiJing to Vaihingen

The second quantitative experiment we conducted is transferring the BEJ dataset to VAI dataset. For POT and VAI, they are both German cities, and have the same spectra. While for both BEJ and VAI, their architectural styles and GSD are quite different, but also they are from different countries and have different spectra. Furthermore, BEJ is collected from satellite sensors but VAI is acquired by airborne sensors. In addition, compared to POT and VAI, BEJ treats both the low vegetation and the tree as the tree, so BEJ dataset only has four classes: impervious surface, building roof, tree and car. During training procedure, we assigned the same label to low vegetation and tree in the VAI dataset.

**Table 2**: The performance of adaptation BEJ to VAI(%), tree denotes low vegetation and tree. The symbol $*$ denotes that the code we implement ourselves.

| | car | roof | tree | road | mIoU |
|---|---|---|---|---|---|
| Souce-only | 2.3 | 30.8 | 0.0 | 44.0 | 19.3 |
| DANN$^*$ [12] | **19.3** | 27.1 | 1.8 | **49.6** | 24.4 |
| ADDA$^*$ [13] | 5.9 | 61.7 | **26.8** | 30.2 | 31.1 |
| MCD-UDA$^*$ [17] | 9.7 | 47.5 | 19.2 | 38.2 | 28.6 |
| Ours | 18.3 | **61.9** | 12.0 | 38.3 | **32.6** |
| Target-only | 60.6 | 85.6 | 88.7 | 79.4 | 78.6 |

The results of this experiment are reported in Table 2, the baseline performance (source-only) is 19.3% and our method achieves a mIoU of 32.6%, thereby improving the baseline by 13.3%. Compared with other competing methods, our approach can yield the best performance, which proves the stability and generalization of our method. The qualitative results are shown in Fig. 4.

## 4. CONCLUSION

In this paper, we present an end-to-end domain adaptation method in the framework of GANs by introducing a novel domain similarity discriminator (DSD) to eliminate the domain shift for semantic segmentation of urban areas. A Siamese network is taken as our domain similarity discriminator to train our network in a pairwise training scheme. By DSD we can use the similar information between the same domain and the dissimilar information between different domains to obtain the domain-invariant features. The experimental results show that our approach can yield the best performance over the competing methods across three different cities.

## 5. REFERENCES

[1] Evan Shelhamer, Jonathan Long, and Trevor Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, vol. 39, no. 4, pp. 640–651, 2017.

[2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.(TPAMI)*, vol. 39, no. 12, pp. 2481–2495, 2017.

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *CoRR*, vol. abs/1412.7062, 2014.

[4] 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," *CoRR*, vol. abs/1606.00915, 2016.

[5] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam, "Rethinking atrous convolution for semantic image segmentation," *CoRR*, vol. abs/1706.05587, 2017.

[6] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan, "A theory of learning from different domains," *Machine Learning (ML)*, vol. 79, no. 1-2, pp. 151–175, 2010.

[7] Sinno Jialin Pan, Ivor W. Tsang, James T. Kwok, and Qiang Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Networks*, vol. 22, no. 2, pp. 199–210, 2011.

[8] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S. Yu, "Transfer feature learning with joint distribution adaptation," in *IEEE International Conference on Computer Vision, (ICCV)*, 2013, pp. 2200–2207.

[9] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan, "Learning transferable features with deep adaptation networks," in *Proceedings of the 32nd International Conference on Machine Learning, (ICML)*, 2015, pp. 97–105.

[10] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *Advances in Neural Information Processing Systems 29 (NIPS)*, 2016, pp. 136–144.

[11] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan, "Deep transfer learning with joint adaptation networks," in *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017, pp. 2208–2217.

[12] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky, "Domain-adversarial training of neural networks," *Journal of Machine Learning Research*, vol. 17, pp. 59:1–59:35, 2016.

[13] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell, "Adversarial discriminative domain adaptation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2962–2971.

[14] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27 (NIPS)*, 2014, pp. 2672–2680.

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

[16] Yang Zhang, Philip David, and Boqing Gong, "Curriculum domain adaptation for semantic segmentation of urban scenes," in *IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2039–2049.

[17] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," *CoRR*, vol. abs/1712.02560, 2017.

[18] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah, "Signature verification using a siamese time delay neural network," in *Advances in Neural Information Processing Systems (NIPS)*, 1993, pp. 737–744.

[19] Akshay Mehrotra and Ambedkar Dukkipati, "Generative adversarial residual pairwise networks for one shot learning," *CoRR*, vol. abs/1703.08033, 2017.

[20] Chris Donahue, Akshay Balsubramani, Julian McAuley, and Zachary C. Lipton, "Semantically decomposing the latent spaces of generative adversarial networks," *International Conference on Learning Representations (ICLR)*, 2018.

[21] Swami Sankaranarayanan, Yogesh Balaji, Arpit Jain, Ser-Nam Lim, and Rama Chellappa, "Unsupervised domain adaptation for semantic segmentation with gans," *CoRR*, vol. abs/1711.06969, 2017.

[22] "International society for photogrammetry and remote sensing 2d semantic labeling challenge," `http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html`.

[23] Facebook Open Source, "Pytorch," `https://github.com/pytorch/pytorch`, 2017.

[24] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 248–255.

[25] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.