Semantic Segmentation Domain Adaptation with Generative Model

Zhanghao Sun, Qiwen Wang, Mi Yu
Stanford University

Abstract

Convolutional neural network-based methods work well on semantic segmentation with labeled data, but may not generalize on different dataset. Since data labeling is expensive, studying methods that can adapt the network learned from one dataset to another is of great interest. Domain adaptation is introduced to solve this problem. In this paper, we study several adversarial models for domain adaptation. Our models are based on a previous network that leverages spatial similarities between source and target domains in output space. We test different ways of fusing the information from multiple network stages. We also propose a new category-level method utilizing a grouped convolutional neural network to mitigate mis-classifications. Extensive experiments are conducted under synthetic-to-real scenarios. We observe that category-level matching provides sharper boundaries than other methods. We also show that an ensemble model out-performances baseline by large margin.

1 Introduction

Semantic segmentation aims to assign each pixel a semantic label such as person, road, or car, in an image. Significant progress has been obtained with methods based on convolutional neural networks (CNNs) using labeled data. However, data labeling is both tedious and labor-intensive, which limits dataset quality and size. As a result, synthetic dataset plays an increasingly important role in training large, robust neural networks.

However, generalization of model trained on synthetic dataset to real-world is challenging because different datasets tend to follow different distributions. Experiments have shown that there are discrepancies between synthetic and real data, commonly known as the "domain gap." For instance, texture, lighting condition and color for synthetic objects may be very different from that of real-world objects. In such cases, relying solely on a supervised model require re-annotating the pixel-level ground truth that entails high labor cost.

To address this issue, "domain adaptation" algorithms are proposed to close the gap between source and target domains. For image classification, one effective approach is to align the feature space of two domains. However, for segmentation problem, using feature adaptation may suffer from the complexity of high-dimensional features that encode various visual cues. This motivates us to investigate methods that instead adapt to the output space. In this work, we wish to improve existing algorithms in this field through several recently proposed ideas on output space alignment. Specifically, we adopt the insights in three previous works [15, 9, 7]. Our contributions are in two folds:

1. We apply the idea of "conditional adversarial domain adaptation" [9] to semantic segmentation task for the first time. This is non-trivial because compared to image classification tasks in the original work [9], image segmentation task deals with much higher dimensional feature space.

2. we propose a new approach for "Category-level domain adaptation" [7, 11, 12] by leveraging the network architecture in [15]. When we apply the adversarial method to the output space, the discriminator can be fooled by similar segmentation with misclassified labels. This motivates us to separate the classes in the convolutional layer of the discriminator.

2 Related Work

2.1 Semantic Segmentation

State-of-the-art semantic segmentation methods are mainly based on recent advances of deep neural networks. As proposed by Long et al. [8], semantic segmentation problem can be tackled by transforming a classification CNN, such as AlexNet, VGG, and ResNet. Our segmentation network uses ResNet101 with enlarging receptive fields [3]. To train this kind of model, annotated labels must be obtained for the images. This is very labor-cost. This is even more problematic as the trained model does not generalize on different images.

2.2 Domain Adaptation

Unsupervised domain adaptation is attracting more and more attention in the computer vision community [2, 15, 1, 17, 16, 13]. The main insight of these approaches is to align the distribution between source and target images, either in the input stage [2], feature stage [10, 6, 13], or output stage [15]. Recently, researches are conducted that challenges several basic assumptions in previous works. First, by combining feature space and output space through a randomized inner product,Long et al. [9] improves the performance of the classification task. Second, by aligning distribution conditioned on semantic class, rather than marginal distribution, is demonstrated to be helpful [12, 11, 7]. This approach is commonly denoted as "Multi-class" domain adaptation in literature. In this work, we integrate these two ideas into the state-of-the-art model [15] to explore their capabilities.

3 Problem Statement

Here we provide a formal statement of our problem. Let $\{x_i\}_{i=1}^N$ with dimension $H \times W \times 3$ denote the images from the input space. Let $\{y_i\}_{i=1}^N$ be the corresponding segmentation label with dimension $H \times W$. H denotes the height of the image, and W denotes the width of the image. We have the source domain $\mathcal{D}_s$ and the target domain $\mathcal{D}_t$ following different distributions. In the scenario of unsupervised domain adaptation, we have access to the source domain labels y_i^s but not the target domain labels y_i^t. Our objective is to close the gap between the source and target domain. In particular, based on "AdaptSegNet" [15], instead of aligning the two domains in the feature space, the model uses discriminator at the output stage, which back-props to align the feature space.

4 Approaches

4.1 Single and Multi-level Adaptation Learning

This network is proposed in the paper AdaptSegnet[15]. This network consists of a ResNet-101 segmentation network, which is pre-trained ResNet-101 [5] on ImageNet classification and a fully convolutional discriminator network. Unlike the image classification problem, which describes the global visual information, high-dimensional features learned from semantic segmentation encodes complex representations. On the other hand, the output space contains rich information such as layout and context of the scene. Therefore, aligning the output space might be a better choice for semantic segmentation. The overall objective of single-level adversarial network can be described as

$$\mathcal{L}(I_s, I_t) = \mathcal{L}_{seg}(I_s) + \lambda_{adv}\mathcal{L}_{adv}(I_t) \tag{1}$$

and the objective for multi-level can be extended to

$$\mathcal{L}(I_s, I_t) = \sum_i \lambda_{seg}^i \mathcal{L}_{seg}^i(I_s) + \sum_i \lambda_{adv}^i \mathcal{L}_{adv}^i(I_t) \tag{2}$$

the λ here are weights used to balance the losses. Given the segmentation softmax output $P = G(I) \in \mathbf{R}^{H \times W \times C}$, where C stands for the number of segmentation labels, the loss for discriminator is a cross-entropy

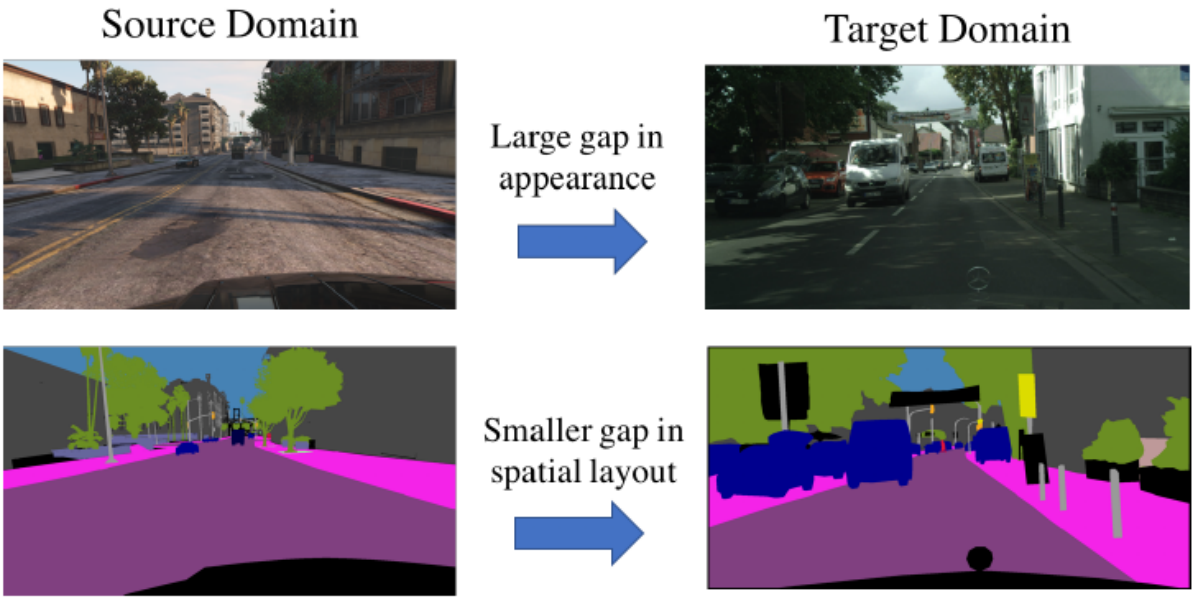

Figure 1: This is an example of our input and label from source domain and target domain. The intuition of alignment on output space is that although the appearance of the images might be very different, the output space of the images can share many similarities.

loss which can be written as

$$\mathcal{L}_d(P) = -\sum_{h,w}(1-z)log(D(P)^{(h,w,0)}) + zlog(D(P)^{(h,w,1)}) \tag{3}$$

where $z = 0$ if the sample is drawn from the target domain and $z = 1$ for the sample from the source domain. Intuitively, the target images can be think of as "fake image" in traditional GAN tasks. The segmentation loss for the source image can be defined as

$$\mathcal{L}_{seg}(I_s) = -\sum_{h,w}\sum_{c\in C} Y_s^{(h,w,c)} log(P_s^{(h,w,c)}) \tag{4}$$

which again is a cross-entropy loss. Here Y_s stands for the ground truth annotations for the source images. The adversarial loss can be written as

$$\mathcal{L}_{adv}(I_t) = -\sum_{h,w} log(D(P_t)^{(h,w,1)}). \tag{5}$$

This loss is designed to train the segmentation network to fool the discriminator by maximizing the probability of thinking this prediction as from source prediction.

4.2 Multi-layer Fusion Network

This idea is inspired by the paper "Conditional Adversarial Domain Adaptation"[9]. The single-level and multi-layer domain adaptation network may fail to recognize the multi-modal structures when the data distribution embodies such complex structures. In other words, even if the discriminator is fully confused, the two distributions are not guaranteed to be sufficiently similar. Second, it is risky to condition the domain discriminator on the discriminative information when it is uncertain. The multi-layer fusion network and category-level network are all aiming to overcome these shortcomings.

The training objective for the semantic segmentation network is the same as above. The only difference here is that when training the discriminator, instead of using one discriminator for each stage, we use various methods to fuse the information together. Intuitively, taking information from different stages together into consideration hopefully can help to capture the modes by realizing the cross-variance dependency between features and classes.
The original paper [9] proposes the model to join distribution using the multi-linear map.

$$T_{\otimes}(f, g) = f \otimes g, \tag{6}$$

where f in our case is the feature map from the second to the last stage with dimension $H \times W \times C$, and g is the output feature map from the last stage with dimension $H \times W \times C$. Comparing to the task of the original paper, the semantic segmentation problem has a significant higher dimension. Therefore, instead of the map used in the original paper, the following equation is used:

$$T(f, g) = f \parallel g. \tag{7}$$

The feature maps f and g are concatenated together and then passed into the discriminator. Concatenating the intermediate feature map and output feature map can help the discriminator to learn some feature-output co-variance dependency.

The second form of the fusing

$$T_{\odot}(f, g) = \frac{1}{\sqrt{d}}(R_f f) \odot (R_g g) \tag{8}$$

is based on the following theorem proved in [9]

Theorem 1 *The expectation and variance of using $T_{\odot}(f, g)$ to approximate $T_{\otimes}(f, g)$ satisfy*

$$\mathbb{E}[< T_{\odot}(f, g), T_{\odot}(f', g') >] =< f, f' >< g, g' >, \tag{9}$$

$$Var[< T_{\odot}(f, g), T_{\odot}(f', g') >] = \sum_{i=1}^{d} \beta(R_i^f)\beta(R_i^g, g) + C, \tag{10}$$

$$\tag{11}$$

where $\beta(R^f, f) = \frac{1}{d}\sum_{j=1}^{d_f}\left[f_j^2 f_j'^2 \mathbb{E}(R_{ij}^f)^4 + C'\right]$ and similarly for $\beta(R_i^g, g)$; C, C' are constants.

In particular, the proof in the original paper shows that such an estimator is an unbiased estimator in terms of the inner product, and its variance depends on the fourth-order moments, which are constant for many symmetric uni-variate distributions. For example, Gaussian distribution has a constant fourth-order moment. Thus this estimator is both unbiased and consistent. Since f and g in semantic segmentation have a very high dimension, constructing randomized matrices R_f and R_g with large dimensions and compute the matrix multiplications impair both the speed and the memory efficiency. In this model, the dimension of the random matrix R^f, R^g is limited to $H \times W \times 50$.

4.3 Category-level Domain Adaptation

Rather than fusing the feature information with the output stage information and use a fully convolutional neural network, we can also use a grouped convolutional neural network instead. A grouped neural network has the following structure:

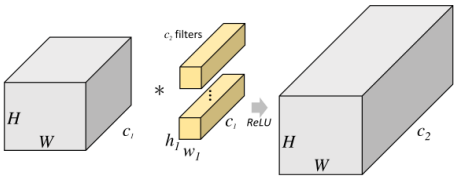

Figure 2: Convolutional layer in the discriminator of the single-level, multi-level and multi-layer fusion models.

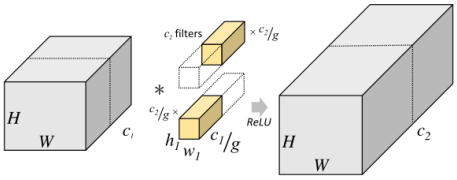

Figure 3: Group convolutions in the discriminator of the category-level model.

Different from the regular convolutional neural network, the grouped convolution does not apply each filter on the entire depth. As the two diagrams above demonstrated, the depth is cut from C to 1, where C denotes the number of classes. We then pass in each grouped information into the discriminator and get C number of adversarial loss. We hope such treatment will effectively fool the discriminator without mismatching the classes.

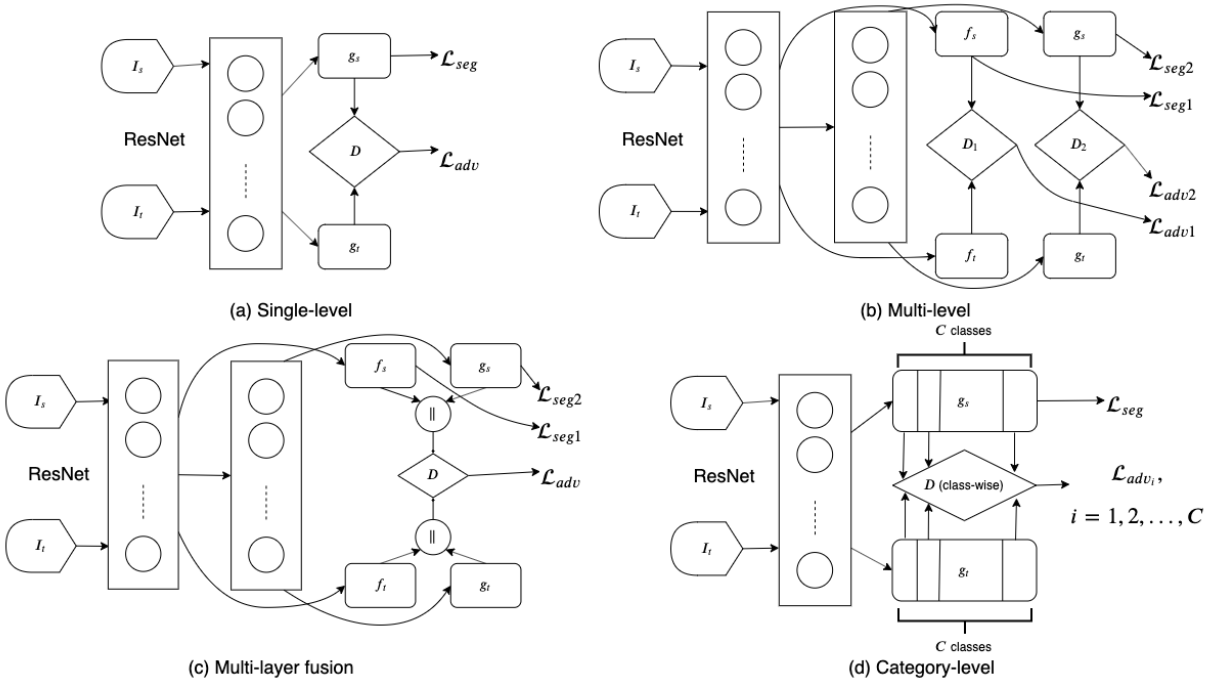

Figure 4: This are the four network structures for the domain adaptation problem. The two in the upper half use discriminator at stage level and compute the loss separately. (c) finds a way to fuse the two stages together (through concatenation or randomized matrices element-wise product.) (d) uses grouped CNN to separate the information before passing into the discriminator.

4.4 Ensemble Method

Ensemble models in machine learning combine the decisions from multiple models in order to achieve an overall better performance. The ensemble model combines the result of our five models including the single level, multi-level, fusion concatenation, fusion random, and categorical models. We have applied 2 ensemble methods: 1) Maximizing: for each pixel and segmentation category, we select the maximum likelihood value among all the models. 2) Averaging: for each pixel and segmentation category, we average the likelihood of all the models. These simple ensemble techniques can usually boast the performance of the model by 1%-2%.

4.5 Extension: Multi-task Domain Adaptation

Although unsupervised domain adaptation has been explored intensively, the scenario of adapting knowledge from one source domain to multiple target domains is seldom studied. However, it is of application value, for example, we can train only one model to adapt GTA5 dataset to two cities with totally different appearances.

To the best of our knowledge, we're the first to conduct experiments in multi-target domain adaptation for semantic segmentation.

In this work, we formulate multi-target domain adaptation problem as an multi-objective optimization. In each training iteration, we get equal number of data from each target dataset. Then, we calculate adversarial loss on both target data. Based on the gradient for these two losses, we calculate weights through "multiple gradient descent algorithm" (MGDA) proposed in [14]. Finally, we back-propagate the weighted loss. We compare our training strategy with a naive training strategy that use equal weights for both loss.

5 Result

In this section we will give a detailed descriptions of the evaluation metrics we used and compare the performance of different models on domain adaptation on semantic segmentation task.

5.1 Dataset

The GTA5 dataset is used as the source domain for our single-task adaptation model. The dataset consists of 24966 images with resolution of 1914×1052 sythesized from the video game GTA5 based on the city of Loas Angeles. The original cityscape dataset [4] contains 5000 images manually selected from 27 real cities with heavy pixel-level annotation. GTA5 dataset ground-truth annotation is compatible with the Cityscapes dataset that contains 19 categories. We use the full GTA5 dataset and adapt the model to the Cityscapes training set with 2975 images. During testing, we evaluate on the Cityscapes validation set with 500 images.

For the multi-target domain adaptation, we add another real-world dataset, named "Crosscity-Taipei". It shares 13 classes with Cityscapes and GTA5 dataset. We use 2975 training data from Taipei and Cityscapes dataset while tested with 100 scenes from both datasets.

5.2 Baseline

Our baseline model will be training the DeepLab-v2 [3] framework with ResNet-101 model pre-trained on ImageNet dataset without any domain adaptation mechanism. The idea behind the baseline is that at least our models need to be better than training without any domain adaptation to demonstrate the significance of domain adaptation mechanism we proposed above.

5.3 Model Evaluation

We use both quantitative and qualitative comparisons for model evaluation.

5.3.1 Qualitative Results

Qualitative comparisons are shown in Figure 5, we selected several representative scenes for discussion. As can be noticed in the figure, all models is better than no adaptation, which means all models to some extent learns to overcome the domain gap between synthetic and real-world scenes. In the first row, all models achieve similar performance, multi-layer discrimination model in [15] resolves most details for the traffic signs and poles. In the second row, building in the background is white and is thus easily be classified as sky. multi-layer discrimination model still achieves best performance in this case, other models are similar, except the category-level model. This model makes intolerance mistake and almost totally classifies background building as sky. We attribute this to the fact that category-level discriminator only have access to one class's segmentation result, thus it lacks "global" respective. Therefore, it just extend the sky region as large as possible, without considering the fact that buildings should not be divide by sky into blocks, as in the result. In the third row, we show the advantage of category-level model that it produces relatively sharp segmentation boundary, which is within our expectation that category-level can separate features from different classes apart further.

5.3.2 Quantitative Results

For quantitative comparisons, we use two metrices, namely, intersection over union (IoU) and dice coefficient. They are defined as below:

$$\text{IoU} = \frac{TP}{TP + FP + FN} \tag{12}$$

$$\text{Dice} = \frac{2TP}{2TP + FP + FN} \tag{13}$$

where TP stands for truth positive, FP stands for false positive, and FN stands for false negative. As can be seen from the formulas, the difference between IoU and dice coefficient is that dice coefficient counts the overlap of ground truth and prediction twice. Thus, when averaging of samples it tends to measure the average performance of the model. On the other hand, IoU exagregates the error and thus is closer to the worst case performance.

Overall and class-wise evaluation over these two metrices are shown in Table 1 & 2. As can be seen from the tables, when comparison is restricted to single model, multi-layer discrimination model in [15] is best. single-layer discrimination model is slightly worse. Our simple concatenation fusion model achieves second-best performance. This is in consistent with the argument in [9], where the concept of randomized inner product is proposed as an improvement from concatenation fusion. We attribute this discrepancy to the fact that semantic segmentation is a much more complicated problem compared to object classification, which is used as demonstration in [9]. Our feature space is much larger than that in . Thus, it requires much larger dimension of the random matrices "R_g" and "R_f" to maintain this information. However, as discussed in Section 4.2, we only use 50 as our feature dimension, due to memory and speed considerations.

Method	road	sidewalk	building	wall	fence	pole	light	sign	veg	terrain	sky	person	rider	car	truck	bus	train	mbike	bike	mIoU
No Adapt	78.4	28.4	74.7	20.4	20.0	21.8	29.2	14.6	77.1	17.5	72.2	55.9	24.2	63.4	20.6	22.8	4.7	24.5	**39.6**	37.4
Single-Level	86.5	25.9	79.8	22.1	20.0	23.6	33.1	**21.8**	81.8	25.9	75.9	57.3	26.2	76.3	29.8	32.1	**7.2**	29.5	32.5	41.4
Multi-Level	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
Fusion Cancat	85.5	28.1	80.4	27.1	23.0	26.0	32.3	21.2	82.9	26.1	74.4	59.0	27.3	**76.9**	34.3	28.7	0.3	29.8	32.4	41.9
Fusion Random	84.4	34.7	77.7	25.6	20.4	24.8	27.8	17.3	81.4	30.1	75.7	57.6	20.8	71.2	27.2	29.7	0.5	28.1	30.1	40.3
Categorical	79.0	30.4	77.5	27.7	20.3	**26.8**	29.9	18.5	80.7	23.2	71.6	56.2	19.9	59.0	25.1	25.4	5.1	22.4	22.3	37.9
Ensemble Max	86.3	32.7	**80.7**	27.9	24.6	26.4	34.7	19.5	**84.2**	32.7	77.4	59.1	27.7	70.7	**36.4**	**48.1**	3.6	32.2	30.1	43.9
Ensemble Avg	**86.9**	33.5	**80.7**	**30.1**	**25.2**	26.6	34.1	19.2	84.1	**33.7**	**77.5**	**59.3**	27.2	72.3	35.5	46.0	3.9	**33.9**	29.1	**44.2**

Table 1: IoU score for each model we applied on each classes as well as the overall IoU score.

Method	road	sidewalk	building	wall	fence	pole	light	sign	veg	terrain	sky	person	rider	car	truck	bus	train	mbike	bike	mdice
No Adapt	87.9	44.3	85.5	33.9	33.4	35.8	45.3	25.5	87.1	29.9	83.9	71.7	39.0	77.7	34.3	37.2	9.2	39.5	**56.8**	50.4
Single-Level	89.8	44.7	**89.6**	38.4	38.9	38.5	**53.1**	32.3	**91.4**	43.6	**88.1**	72.3	**44.6**	72.7	46.2	59.4	10.8	46.1	31.3	54.3
Multi-Level	91.7	44.3	87.7	42.5	36.2	40.7	50.6	28.9	91.0	44.6	87.1	73.0	35.8	86.4	52.2	**66.9**	**15.6**	50.5	34.8	55.8
Fusion Cancat	92.2	44.0	89.2	42.7	37.5	41.3	48.9	**35.1**	90.7	41.5	85.3	74.3	43.0	**87.0**	51.2	44.7	0.8	46.0	48.9	55.0
Fusion Random	91.5	**51.6**	87.5	40.9	34.0	39.8	43.5	29.6	89.8	46.3	86.2	73.1	34.5	83.2	42.8	45.9	1.0	43.9	46.3	53.2
Categorical	88.3	46.7	87.3	43.4	33.8	**42.3**	46.1	31.3	89.3	37.7	83.5	72.0	33.3	74.2	40.2	40.6	9.8	36.6	36.5	51.2
Ensemble Max	92.6	49.2	89.3	43.7	39.5	41.7	51.5	32.7	**91.4**	49.2	**87.3**	74.3	43.4	82.8	**53.0**	64.9	6.9	48.8	46.3	57.3
Ensemble Avg	**93.0**	50.2	89.3	**46.3**	**40.3**	42.1	50.9	32.3	**91.4**	**50.4**	**87.3**	**74.5**	42.8	84.2	52.4	63.0	7.5	**50.6**	45.1	**57.6**

Table 2: Dice coefficient for each model we applied on each classes as well as the overall Dice coefficient.

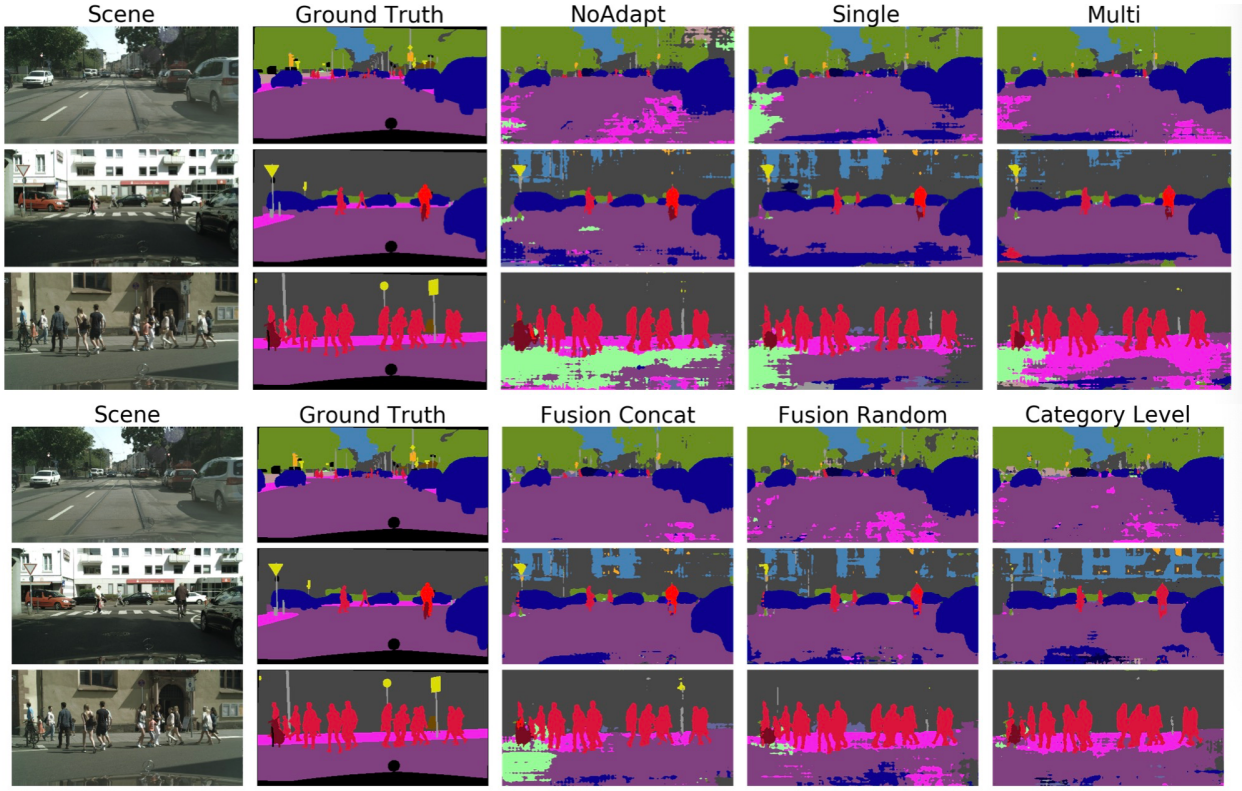

Figure 5: These are the semantic labels produced from our models using three images selected from Cityscapes dataset in our validation dataset.

Finally, the ensemble model achieves best performance by large margin, which indicates improvement introduced by our proposed models is significant. The ensemble averaging performs better than the ensemble maximizing; while the ensemble maximizing may select the category from the sub-optimal model with the highest likelihood, the ensemble averaging considers the likelihood from all the model. Overall, the ensemble method boasts the IoU by 4.2% and the dice coefficient by 1.9%, although in the single model comparisons it might be overwhelmed by some intrinsic draw-backs.

5.4 Multi-target domain adaptation

Due to limited time-scale, we are unable to get proper MGDA training result. We use single-layer model from [15] as our network. Also, since we need to train on two batches, one from each target dataset, memory consumption is around two times larger, so we use VGG16 as our generator, instead of ResNet 108. However, model trained with naive training strategy (Section 4.5) got an IoU of 34.3% on the new "Taipei" dataset, which is quite low. Thus, it motivates a better training strategy. Further experiments are needed to analysis these data and improve multi-target training strategy.

6 Conclusion and Future Work

In conclusion, we established three models for semantic segmentation domain adaptation. Through comparisons with previous state-of-the-art model, we explored two aspects of the network architecture: First, how to effectively combine features from multiple network stages to facilitate the discrimination between different domains. Second, how to introduce category-level discrimination into domain-level discrimination for better class clustering. Our proposed method show certain advantages over previous ones. We also proposed a multi-target domain adaptation model based on multi-objective optimization.

For future work, it would be very intriguing to explore more on the multi-target domain adaptation task. Also, as can be seen in Table 6 and Figure 5, class imbalance is quite obvious in both synthetic and real-world datasets. Solving this problem would be quite useful for semantic segmentation task.

References

[1] Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, and Mario Marchand. Domain-Adversarial Neural Networks. *arXiv e-prints*, page arXiv:1412.4446, Dec 2014.

[2] Amir Atapour-Abarghouei and Toby P Breckon. Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2800–2810, 2018.

[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. *arXiv e-prints*, page arXiv:1606.00915, Jun 2016.

[4] 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. *arXiv e-prints*, page arXiv:1604.01685, Apr 2016.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[6] Weixiang Hong, Zhenzhen Wang, Ming Yang, and Junsong Yuan. Conditional generative adversarial network for structured domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1335–1344, 2018.

[7] Abhishek Kumar, Prasanna Sattigeri, Kahini Wadhawan, Leonid Karlinsky, Rogerio Feris, Bill Freeman, and Gregory Wornell. Co-regularized alignment for unsupervised domain adaptation. In *Advances in Neural Information Processing Systems*, pages 9345–9356, 2018.

[8] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully Convolutional Networks for Semantic Segmentation. *arXiv e-prints*, page arXiv:1411.4038, Nov 2014.

[9] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, pages 1640–1650, 2018.

[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*, pages 136–144, 2016.

[11] Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2507–2516, 2019.

[12] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3723–3732, 2018.

[13] Swami Sankaranarayanan, Yogesh Balaji, Carlos D Castillo, and Rama Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8503–8512, 2018.

[14] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In *Advances in Neural Information Processing Systems*, pages 527–538, 2018.

[15] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7472–7481, 2018.

[16] Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. Person transfer gan to bridge domain gap for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 79–88, 2018.

[17] Kaichao You, Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan. Universal domain adaptation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.