Consistency Regularization for Domain Adaptation

Kian Boon Koh[1,2] and Basura Fernando[1,2][0000−0002−6920−9916]

[1] Institute of High Performance Computing, A*STAR, Singapore
[2] Centre for Frontier AI Research, A*STAR, Singapore
kianboonkoh@gmail.com
fernando_basura@ihpc.a-star.edu.sg

Abstract. Collection of real world annotations for training semantic segmentation models is an expensive process. Unsupervised domain adaptation (UDA) tries to solve this problem by studying how more accessible data such as synthetic data can be used to train and adapt models to real world images without requiring their annotations. Recent UDA methods applies self-learning by training on pixel-wise classification loss using a student and teacher network. In this paper, we propose the addition of a consistency regularization term to semi-supervised UDA by modelling the inter-pixel relationship between elements in networks' output. We demonstrate the effectiveness of the proposed consistency regularization term by applying it to the state-of-the-art DAFormer framework and improving mIoU19 performance on the GTA5 to Cityscapes benchmark by 0.8 and mIou16 performance on the SYNTHIA to Cityscapes benchmark by 1.2.

1 Introduction

Semantic segmentation is a task which requires a lot of pixel level annotations and obtaining these annotations is expensive and time consuming. To overcome this issue, one solution is to obtain annotations from synthetic data such as Games (e.g. GTA5) and train models on these synthetic data for semantic segmentation. However, the problem is that even if modern synthetic data is near photo realistic, still there is a distribution mismatch between the synthetic data and real images. One solution is to develop models that can overcome this distribution mismatch between models that are trained on synthetic data and real data which is the topic of unsupervised domain adaptation (UDA) [7,8,13,10,22,2,29,9].

UDA for semantic segmentation has made significant progress in recent years. One of the most recent method called DAFormer [16] obtained massive improvement over prior methods by using a Transformer architecture and self-training. However one of the challenges in self-training is that generated pseudo labels can be wrong and that may result in poor transfer of information from source domain to the target domain. Therefore, it is needed to further regularize the self-training learning process.

In this work, we present a new consistency regularization method based on correlation between pixel-wise class predictions. We enforce two models (teacher and student) to have similar inter-pixel similarity structure and by doing so we regularize the self-training process. This helps to improve the generalization of the student network as well as the teacher network allowing better transfer of information from the source domain to the target domain. We demonstrate its effectiveness by applying it to DAFormer and improving mIoU19 performance on the GTA5 to Cityscapes benchmark by 0.8 and mIou16 performance on the SYNTHIA to Cityscapes benchmark by 1.2. Implementation of our proposed method is available at our GitHub repository[3].

2 Related Work

2.1 Unsupervised Domain Adaptation

Domain adaptation is a field of techniques that aims to solve the domain shift problem, when data distributions experience change between datasets. UDA is a subset of the domain adaptation field that aims to utilize a labeled source domain to learn a model that performs well on an unlabeled target domain. Recent UDA methods can be grouped into either adversarial training or self-supervised learning (SSL) approaches. Adversarial training methods aim to reduce source and target distribution mismatch by aligning distributions at either the pixel [14,11,3] or intermediate feature level [30,15] using a generative adversarial network (GAN).

SSL methods allow models to be trained directly on the target domain by generating pseudo labels from the target domain. Recent advances focuses on improving the quality of pseudo labels using various approaches, such as using representative prototypes [37] or using more complex, Transformer-based network architecture [16]. It is also possible for methods to adopt a hybrid approach and use both adversarial training and SSL. Li et al. does so in their bidirectional learning framework [18]. Adversarial training is first used to obtain an image-to-image translation model and a segmentation model. Target domain pseudo labels are then generated from high confidence predictions, which are then subsequently used to fine tune the segmentation model. The improved segmentation model can then be used in the first adversarial stage to form a close loop.

2.2 Semantic Segmentation

Early methods on semantic segmentation problems were largely based on Fully Convolutional Network (FCN) [26], which typically follows an encoder-decoder architecture [1,24]. Further improvements were made by using dilated convolutions to overcome the loss of spatial resolution [34], and pyramid pooling [38,4] to enhance capturing of contextual information. Recent success of attention-based

[3] `https://github.com/kw01sg/CRDA`

Transformers [31] in natural language processing has seen adaptations of Transformers for image segmentation [19,33] that were able to obtain state-of-the-art results.

2.3 Consistency Regularization

Consistency regularization is a regularization technique used to encourage networks to make consistent predictions that are invariant to perturbations. Tarvainen and Valpola improved model performance on the image classification problem by using a student and teacher network pair in their Mean Teacher model [28], where the weights of the teacher network are an exponential moving average (EMA) of the student network. Consistent predictions between the two networks are then promoted by optimizing a consistency loss between their predictions. Interpolation consistency training by Verma et al. [32] combines mixup [36] and the Mean Teacher model [28] to implement consistency regularization. During training, unlabelled samples are interpolated to create an augmented sample. Predictions by the student network on the augmented sample are then optimized to be consistent with interpolated predictions by the teacher network on the original non-interpolated samples. Kim et al. [17] uses cosine similarity in their consistency regularization method for semantic segmentation. They propose a structured consistency loss that optimizes predictions to be consistent in not only pixel-wise classification, but also inter-pixel relationship.

3 Our Method

Given source domain images $x_S \in X_S$ with their annotations (labels) $y_S \in Y_S$ and target domain images $x_T \in X_T$ without annotations (labels), we want to learn a network h that can correctly predict the annotations for target images X_T denoted by $\hat{Y_T}$. Typically, there is a mismatch in the joint probability distributions of source domain data $P(X_S, Y_S)$ and the target domain data $P(X_T, Y_T)$. Due to this mismatch or the gap between source and target domains, an image segmentation model h that is trained on the source data usually results in a low performance on target images. One common solution to address this issue is to use self-training as also done in the prior works such as DAFormer [16]. However, semi-supervised self-training methods could easily over-fit to source distribution and could generate inconsistent or wrong pseudo labels for the target domain images. To overcome this limitation, we propose the addition of consistency regularization to the DAFormer [16] framework during model training to further improve model performance. Next, we explain the overall training framework.

3.1 Overall Training

Overall training of the network is composed of three components: supervised training using source images, self-training using target images, and consistency regularization. Total loss $\mathcal{L}_{total}$ is given as

$$\mathcal{L}_{total} = \mathcal{L}_S + \mathcal{L}_T + \lambda_c \mathcal{L}_C \tag{1}$$

where $\mathcal{L}_S$ is supervised cross entropy loss using source images, $\mathcal{L}_T$ is self-trained cross entropy loss using pseudo labels, $\mathcal{L}_C$ is our consistency regularization term, and λ_c is a parameter we use to weigh $\mathcal{L}_C$. The following sections will present each of the losses in detail.

Supervised Training Supervised training on the source domain is conducted using cross entropy loss for semantic segmentation. For a source image x_S and its annotation y_S, $\mathcal{L}_S$ can be defined as

$$\mathcal{L}_S(x_S, y_S) = -\frac{1}{HW} \sum_{j=1}^{H \times W} \sum_{c=1}^{C} y_S^{(j,c)} \log h(x_S)^{(j,c)} \tag{2}$$

where C is the number of classes and H and W are the height and width of the segmentation output. The notation $y_S^{(j,c)}$ denotes the presence of class c at pixel location j (1 if present and 0 if not). Similarly, $h(x_S)^{(j,c)}$ denotes the predicted score for class c at pixel location j using model h for image x_S.

Self-Training Self-training uses a teacher network $f(;\phi)$ to produce pseudo labels on which the student network $h(;\theta)$ will be trained on. For a target image x_T, its pseudo label p_T is formally defined as

$$p_T^{(j,c)} = [\![c = \arg\max_{c'} f(x_T;\phi)^{(j,c')}]\!] \tag{3}$$

where $[\![\cdot]\!]$ denotes the Iverson bracket.

We follow the Mean Teacher model [28] where the weights of the teacher network $f(;\phi)$ are the EMA of the weights of the student network $h(;\theta)$ after each training step t. The EMA weights used by the teacher model at training step t is formally defined as

$$\phi_{t+1} = \alpha\phi_t + (1-\alpha)\theta_t \tag{4}$$

where ϕ_{t+1} is the EMA of successive weights and α is a smoothing coefficient hyperparameter. It should also be noted that no gradients will be backpropagated into the teacher network from the student network.

A confidence estimate for the pseudo labels, defined as the ratio of pixels with maximum softmax probability exceeding a pre-defined threshold τ, is also used in the self-training loss. For a target image x_T, its confidence estimate q_T is formally defined as

$$q_T = \frac{\sum_{j=1}^{H \times W} [\max_{c'} f(x_T;\phi)^{j,c'} > \tau]}{HW} \tag{5}$$

Self-training loss of the student network $\mathcal{L}_T$ for a target image x_T can thus be defined as

$$\mathcal{L}_T(x_T) = -\frac{1}{HW} \sum_{j=1}^{H \times W} \sum_{c=1}^{C} q_T \times p_T^{(j,c)} \times \log h(x_T; \theta)^{(j,c)} \tag{6}$$

We follow DAFormer's [16] method of using non-augmented target images for the teacher network f to generate pseudo labels and augmented targeted images to train the student network h using Equation 6. We also follow their usage of color jitter, Gaussian blur, and ClassMix [21] as data augmentations in our training process.

Consistency Regularization As mentioned in Mean Teacher [28], cross entropy loss in Equation 6 between predictions of the student model and pseudo labels (which are predictions from the teacher model) can be considered as a form of consistency regularization. However, different from classification problems, semantic segmentation problems have a property where pixel-wise class predictions are correlated with each other. Thus, we propose to further enhance consistency regularization by focusing on this inter-pixel relationship. Inspired by the method of Kim et al. [17], we use the inter-pixel cosine similarity of networks' predictions on target images to regularize the model. Formally, we define the similarity between pixel i and j class predictions on a target image x_T as

$$a_{i,j} = \frac{\mathbf{p}_i^T \mathbf{p}_j}{\|\mathbf{p}_i\| \cdot \|\mathbf{p}_j\|} \tag{7}$$

where $a_{i,j}$ represents the cosine similarity between the prediction vector of the ith pixel and the prediction vector of the jth pixel. Note that the similarity between the probability vector $\mathbf{p}_i$ and $\mathbf{p}_j$ can also be computed using Kullback-Leibler (KL) divergence and cross entropy. We investigate these options in Section 4.3. The consistency regularization term, $\mathcal{L}_C$ can then be defined as the mean squared error (MSE) between the student network's similarity matrix and the teacher network's similarity matrix

$$\mathcal{L}_C = \frac{1}{(HW)^2} \sum_{i=1}^{H \times W} \sum_{j=1}^{H \times W} \left\| a_{i,j}^s - a_{i,j}^t \right\|^2 \tag{8}$$

where $a_{i,j}^s$ is the similarity obtained from the student network and $a_{i,j}^t$ is the similarity obtained from the teacher network. We also follow the method of Kim et al. [17] to restrict the number of pixels used in the calculation of similarity matrices by performing a random sample of N_{pair} pixels for comparison. Thus, the consistency regularization in Equation 8 is updated to the following equation

$$\mathcal{L}_C = \frac{1}{(N_{pair})^2} \sum_{i=1}^{N_{pair}} \sum_{j=1}^{N_{pair}} \left\| a_{i,j}^s - a_{i,j}^t \right\|^2 \tag{9}$$

This term $\mathcal{L}_C$ is particularly useful for domain adaptation as it helps to minimize the divergence between the source representation and the target representation by enforcing a structural consistency in the image segmentation task.

4 Experiments

4.1 Implementation Details

Datasets We use the Cityscapes street scenes dataset [6] as our target domain. Cityscapes contains 2975 training and 500 validation images with resolution of 2048×1024, and is labelled with 19 classes. For our source domain, we use the synthetic datasets GTA5 [23] and SYNTHIA [25]. GTA5 contains 24,966 images with resolution of 1914×1052, and is labelled with the same 19 classes as Cityscapes. For compatibility, we use a variant of SYNTHIA that is labelled with 16 of the 19 Cityscapes classes. It contains 9,400 images with resolution of 1280×760. Following DAFormer [16], we resize images from Cityscapes to 1024×512 pixels and images from GTA5 to 1280×720 pixels before training.

Network Architecture Our implementation is based on DAFormer [16]. Previous UDA methods mostly used DeepLabV2 [5] or FCN8s [26] network architecture with ResNet [12] or VGG [27] backbone as their segmentation model. DAFormer proposes an updated UDA network architecture based on Transformers that was able to achieve state-of-the-art performance. They hypothesized that self-attention is more effective than convolutions in fostering the learning of domain-invariant features.

Training We follow DAFormer [16] and train the network with AdamW [20], a learning rate of $\eta_{base} = 6 \times 10^{-5}$ for the encoder and 6×10^{-4} for the decoder, a weight decay of 0.01, linear learning rate warmup with $t_{warm} = 1500$, and linear decay. Images are randomly cropped to 512×512 and trained for 40,000 iterations on a batch size of 2 on a NVIDIA GeForce RTX 3090. We also adopt DAFormer's training strategy of rare class sampling and thing-class ImageNet feature distance to further improve results. For hyperparameters used in self-training, we follow DAFormer and set $\alpha = 0.99$ and $\tau = 0.968$. For hyperparameters used in consistency regularization, we set $N_{pair} = 512$, $\lambda_c = 1.0$ when calculating similarity using cosine similarity and $\lambda_c = 0.8 \times 10^{-3}$ when calculating similarity using KL divergence.

4.2 Results

Table 1. Comparison with other UDA methods on GTA5 to Cityscapes. Results for DAFormer and our method using cosine similarity are averaged over 6 random runs, while results for our method using KL Divergence are averaged over 3 random runs

Method	Road	Sidewalk	Build.	Wall	Fence	Pole	Tr.Light	Sign	Veget.	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	M.bike	Bike	mIoU19
BDL	91.0	44.7	84.2	34.6	27.6	30.2	36.0	36.0	85.0	43.6	83.0	58.6	31.6	83.3	35.3	49.7	3.3	28.8	35.6	48.5
ProDA	87.8	56.0	79.7	46.3	44.8	45.6	53.5	53.5	88.6	45.2	82.1	70.7	39.2	88.8	45.5	59.4	1.0	48.9	56.4	57.5
DAFormer	95.5	68.9	89.3	**53.2**	49.3	47.8	**55.5**	61.2	89.5	47.7	91.6	71.1	43.3	91.3	67.5	77.6	65.5	53.6	61.2	67.4
Ours (Cosine)	95.5	69.2	**89.5**	52.1	**49.6**	48.9	55.2	**62.1**	89.8	49.0	91.1	**71.7**	**45.1**	**91.7**	**70.0**	77.6	65.2	**56.6**	62.8	68.0
Ours (KL)	**96.1**	**71.6**	**89.5**	**53.2**	48.6	**49.5**	54.7	61.1	**90.0**	**49.4**	**91.7**	70.7	44.0	91.6	**70.0**	**78.1**	**68.9**	55.1	**62.9**	**68.2**

Table 2. Comparison with other UDA methods on SYNTHIA to Cityscapes. Results for DAFormer and our method using cosine similarity are averaged over 6 random runs, while results for our method using KL Divergence are averaged over 3 random runs

Method	Road	Sidewalk	Build.	Wall	Fence	Pole	Tr.Light	Sign	Veget.	Sky	Person	Rider	Car	Bus	M.bike	Bike	mIoU16	mIoU13
BDL	86.0	46.7	80.3	-	-	-	14.1	11.6	79.2	81.3	54.1	27.9	73.7	42.2	25.7	45.3	-	51.4
ProDA	87.8	45.7	84.6	37.1	0.6	44.0	54.6	37.0	88.1	84.4	74.2	24.3	88.2	51.1	40.5	40.5	55.5	62.0
DAFormer	80.5	37.6	87.9	**40.3**	**9.1**	**49.9**	**55.0**	51.8	85.9	88.4	73.7	**47.3**	87.1	58.1	53.0	61.0	60.4	66.7
Ours (Cosine)	86.3	44.2	**88.3**	39.2	7.5	49.2	54.7	**54.7**	**87.2**	**90.7**	**73.8**	**47.3**	**87.4**	55.9	53.7	60.7	61.3	68.1
Ours (KL)	**89.0**	**49.6**	88.1	**40.3**	7.3	49.2	53.5	52.1	87.0	88.0	**73.8**	46.4	87.1	**58.7**	**53.9**	**61.7**	**61.6**	**68.4**

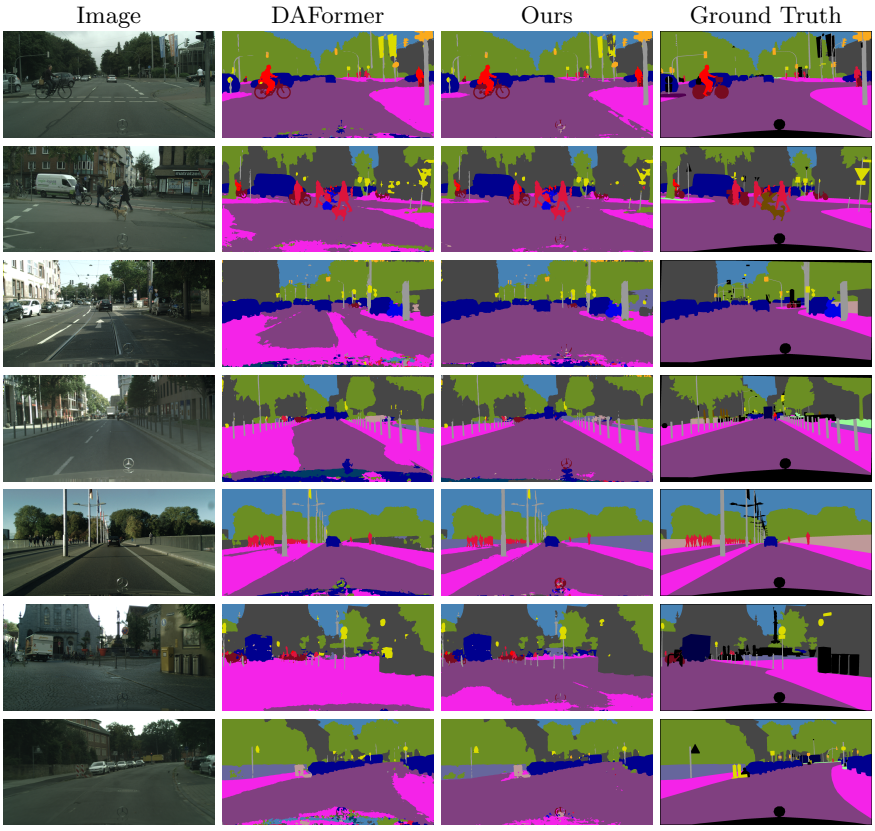


Fig. 1. Qualitative results comparing predictions on validation data of Cityscapes. From left: input image, predictions by DAFormer, predictions by our method, and the ground truth. The last row provides an example where DAFormer performed better compared to our method as it was able to correctly predict the sidewalks

We compare the results of our method against other state-of-the-art UDA segmentation methods such as BDL [18], ProDA [37] and DAFormer [16]. In table 1, we present our experimental results on the GTA5 to Cityscapes problem. It can be observed that our method improved UDA performance from an mIoU19 of 67.4 to 68.0 when cosine similarity is used in Equation 7 and 68.2 when KL divergence is used. Table 2 shows our experimental results on the SYNTHIA to Cityscapes problem. Similarly, our method improved performance from an mIoU16 of 60.4 to 61.3 with cosine similarity and 61.6 with KL divergence.

We also observed that our method was able to make notable improvements on the "Road" and "Sidewalk" categories. This is especially so on the SYNTHIA to Cityscapes problem, where we improved UDA performance on "Road" from 80.5 to 89.0 and "Sidewalk" from 37.6 to 49.6. We further verify this improvement in our qualitative analysis presented in Figure 1, where we observed that our method had better recognition on the "Sidewalk" and "Road" categories. We attribute this improvement to our method's effectiveness in generating more accurate pseudo labels. We present pseudo labels generated during the training process in Figure 2, where we observed more accurate pseudo labels for the "Road" and "Sidewalk" categories.

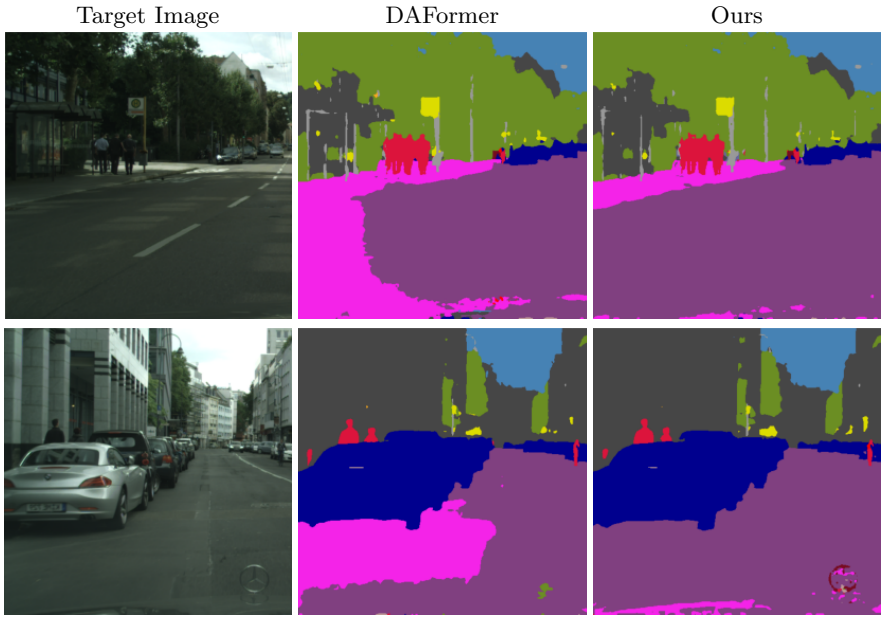


Fig. 2. Qualitative results on pseudo labels generated from the training data of Cityscapes. From left: target image, pseudo labels generated by DAFormer, and pseudo labels generated by our method using cosine similarity

It should be noted that experimental results obtained using the DAFormer method in Tables 1 and 2 were obtained by averaging 6 random runs using the official DAFormer implementation[4]. Even though we were unable to reproduce the exact numbers published in the DAFormer paper, we believe our experimental results for DAFormer are comparable.

4.3 Ablation Study

Table 3. Influence of N_{pair} on UDA performance. Results for all experiments were averaged over 3 random runs except for $N_{pair} = 512$, which was an average over 6 runs

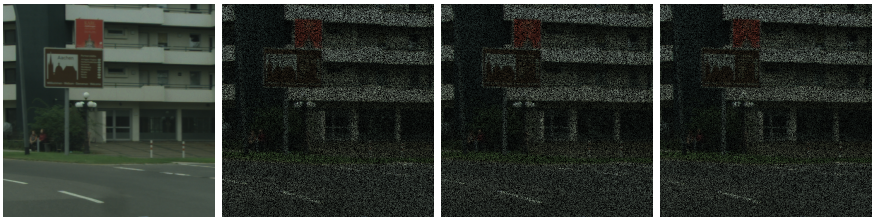
N_{pair}	4	16	64	256	512	1024
mIoU16	**61.4**	60.8	**61.4**	61.1	61.3	60.6

Number of Pixels Sampled We conducted additional experiments on SYNTHIA to Cityscapes to observe the effect of N_{pair} (from Equation 9) on model performance. Theoretically, sampling more pixels for similarity calculation (i.e. a larger N_{pair}) allows us to have a more complete model of the inter-pixel relationship between predictions. However, empirical results in table 3 suggests that N_{pair} does not have significant influence on UDA performance. We observe that very small samples, such as $N_{pair} = 4$, were able to obtain comparable results with larger sample sizes.

Additional experiments using $N_{pair} = 4$ were conducted to observe the locations of sampled pixels. Visualization of our ablation study is presented in Figure 3. We found that after 40,000 training iterations, sampled pixels covered approximately 45.73% of the 512×512 images the network was trained on despite the small sampling size. This suggests that if a reasonable image coverage can be obtained during the training process, a small N_{pair} is sufficient to model the inter-pixel relationship between predictions, allowing us to minimize computational cost of our consistency regularization method. The influence of sampling coverage and sampling distribution on the effectiveness of consistency regularization is an interesting study that can be explored in the future.

Proximity of Sampled Pixels Kim et al. adopted cutmix augmentation [35] in their consistency regularization method [17] to limit sampled pixel pairs to within a local region. They theorized that pixel pairs that are in close proximity to each other have high correlation, and hence have more effect on UDA performance. We tested this theory on SYNTHIA to Cityscapes by performing N_{box} crops and sampling N_{pair} pixels from each crop. This localizes sampled pixels and restricts

[4] `https://github.com/lhoyer/DAFormer`



(a) (b) (c) (d)

Fig. 3. Visualization of pixels sampled in experiments using $N_{pair} = 4$. (a) Target image cropped to 512×512; (b), (c) and (d) visualizes sampled pixels in three separate runs

them to have closer proximity. Sampled pixels are then used to compute inter-pixel similarity to obtain a $N_{box} \times N_{pair} \times N_{pair}$ similarity matrix which is used for loss calculation in Equation 8. We present the experimental results in Table 4

Table 4. Influence of N_{box} and N_{pair} on UDA performance. Total number of sampled pixels i.e. $N_{box} \times N_{pair}$ is kept at 512 for a fair comparison

Crop Size	N_{box}	N_{pair}	mIoU16
256	32	16	**61.2**
128	32	16	61.1
64	32	16	60.2

where three different crop sizes were varied to restrict the proximity of sampled pixels. We did not observe an improvement in UDA performance compared to results presented in Table 2, suggesting that proximity of sampled pixels perhaps may not be that influential for consistency regularization.

Measuring Inter-Pixel Similarity In Section 3.1, we adopted the method of Kim et al. to use cosine similarity in the measure of inter-pixel similarity [17]. In this section, we conducted additional experiments on SYNTHIA to Cityscapes to observe the influence different methods of measuring inter-pixel similarity have on UDA performance.

Table 5. Comparison of UDA performance using different methods to calculate inter-pixel similarity. We also provide the optimal λ_c obtained using hyperparameter tuning

Method	λ_c	mIoU16
Cosine Similarity	1.0	61.3
Cross Entropy	1.0×10^{-3}	61.2
KL Divergence	0.8×10^{-3}	**61.6**

We tested the usage of cross entropy and KL divergence to measure inter-pixel similarity instead of cosine similarity in Equation 7. Results from our empirical experiments are presented in Table 5. We observed that all three methods provided comparable results with each other, with KL divergence providing slightly better results.

5 Conclusion

In this work we presented a new consistency regularization method for UDA based on relationships between pixel-wise class predictions from semantic segmentation models. Using this technique we were able to improve the performance of the state-of-the-art DAFormer method. We also observed that even with smaller number of sampled pixel pairs N_{pair}, this regularization method was still able to be effective. Therefore, with minimal computational cost, we are able to improve the results of self-training methods for unsupervised domain adaptation.

Acknowledgment This research is supported by the Centre for Frontier AI Research (CFAR) and Robotics-HTPO seed fund C211518008.

References

1. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**, 2481–2495 (2017)
2. Ben-David, S., Blitzer, J., Crammer, K., Pereira, F.: Analysis of representations for domain adaptation. Advances in neural information processing systems **19** (2006)
3. Bousmalis, K., Silberman, N., Dohan, D., Erhan, D., Krishnan, D.: Unsupervised pixel-level domain adaptation with generative adversarial networks. CoRR **abs/1612.05424** (2016)
4. Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Transactions on Pattern Analysis & Machine Intelligence **40**(04), 834–848 (apr 2018). https://doi.org/10.1109/TPAMI.2017.2699184
5. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Transactions on Pattern Analysis and Machine Intelligence **PP** (06 2016). https://doi.org/10.1109/TPAMI.2017.2699184
6. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
7. Fernando, B., Aljundi, R., Emonet, R., Habrard, A., Sebban, M., Tuytelaars, T.: Unsupervised Domain Adaptation Based on Subspace Alignment, pp. 81–94. Springer International Publishing, Cham (2017)
8. Fernando, B., Habrard, A., Sebban, M., Tuytelaars, T.: Unsupervised visual domain adaptation using subspace alignment. In: Proceedings of the IEEE international conference on computer vision. pp. 2960–2967 (2013)

9. Fernando, B., Habrard, A., Sebban, M., Tuytelaars, T.: Subspace alignment for domain adaptation. CoRR **abs/1409.5241** (2014), http://arxiv.org/abs/1409.5241
10. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: International conference on machine learning. pp. 1180–1189. PMLR (2015)
11. Gong, R., Li, W., Chen, Y., Van Gool, L.: Dlow: Domain flow for adaptation and generalization. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2472–2481 (2019). https://doi.org/10.1109/CVPR.2019.00258
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90
13. Herath, S., Harandi, M., Fernando, B., Nock, R.: Min-max statistical alignment for transfer learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9288–9297 (2019)
14. Hoffman, J., Tzeng, E., Park, T., Zhu, J.Y., Isola, P., Saenko, K., Efros, A., Darrell, T.: CyCADA: Cycle-consistent adversarial domain adaptation. In: Dy, J., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 80, pp. 1989–1998. PMLR (10–15 Jul 2018), https://proceedings.mlr.press/v80/hoffman18a.html
15. Hoffman, J., Wang, D., Yu, F., Darrell, T.: Fcns in the wild: Pixel-level adversarial and constraint-based adaptation (2016)
16. Hoyer, L., Dai, D., Gool, L.V.: Daformer: Improving network architectures and training strategies for domain-adaptive semantic segmentation. CoRR **abs/2111.14887** (2021), https://arxiv.org/abs/2111.14887
17. Kim, J., Jang, J., Park, H.: Structured consistency loss for semi-supervised semantic segmentation. CoRR **abs/2001.04647** (2020), https://arxiv.org/abs/2001.04647
18. Li, Y., Yuan, L., Vasconcelos, N.: Bidirectional learning for domain adaptation of semantic segmentation. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6929–6938 (2019). https://doi.org/10.1109/CVPR.2019.00710
19. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
20. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=Bkg6RiCqY7
21. Olsson, V., Tranheden, W., Pinto, J., Svensson, L.: Classmix: Segmentation-based data augmentation for semi-supervised learning. In: 2021 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 1368–1377 (2021). https://doi.org/10.1109/WACV48630.2021.00141
22. Pan, S.J., Tsang, I.W., Kwok, J.T., Yang, Q.: Domain adaptation via transfer component analysis. IEEE transactions on neural networks **22**(2), 199–210 (2010)
23. Richter, S.R., Vineet, V., Roth, S., Koltun, V.: Playing for data: Ground truth from computer games. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) European Conference on Computer Vision (ECCV). LNCS, vol. 9906, pp. 102–118. Springer International Publishing (2016)
24. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F.

(eds.) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. pp. 234–241. Springer International Publishing, Cham (2015)
25. Ros, G., Sellart, L., Materzynska, J., Vazquez, D., Lopez, A.M.: The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3234–3243 (2016). https://doi.org/10.1109/CVPR.2016.352
26. Shelhamer, E., Long, J., Darrell, T.: Fully convolutional networks for semantic segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(4), 640–651 (2017). https://doi.org/10.1109/TPAMI.2016.2572683
27. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv 1409.1556 (09 2014)
28. Tarvainen, A., Valpola, H.: Weight-averaged consistency targets improve semi-supervised deep learning results. CoRR **abs/1703.01780** (2017)
29. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7167–7176 (2017)
30. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. CoRR **abs/1702.05464** (2017)
31. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017), https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
32. Verma, V., Lamb, A., Kannala, J., Bengio, Y., Lopez-Paz, D.: Interpolation consistency training for semi-supervised learning. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19. pp. 3635–3641. International Joint Conferences on Artificial Intelligence Organization (7 2019). https://doi.org/10.24963/ijcai.2019/504, https://doi.org/10.24963/ijcai.2019/504
33. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: Segformer: Simple and efficient design for semantic segmentation with transformers. arXiv preprint arXiv:2105.15203 (2021)
34. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. In: ICLR (2016)
35. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: Cutmix: Regularization strategy to train strong classifiers with localizable features. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (October 2019)
36. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. In: International Conference on Learning Representations (2018), https://openreview.net/forum?id=r1Ddp1-Rb
37. Zhang, P., Zhang, B., Zhang, T., Chen, D., Wang, Y., Wen, F.: Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. arXiv preprint arXiv:2101.10979 (2021)
38. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6230–6239 (2017). https://doi.org/10.1109/CVPR.2017.660