Learning Texture Invariant Representation for Domain Adaptation of Semantic Segmentation

Myeongjin Kim Hyeran Byun
Yonsei University
{myeongjin.kim,hrbyun}@yonsei.ac.kr

Abstract

Since annotating pixel-level labels for semantic segmentation is laborious, leveraging synthetic data is an attractive solution. However, due to the domain gap between synthetic domain and real domain, it is challenging for a model trained with synthetic data to generalize to real data. In this paper, considering the fundamental difference between the two domains as the texture, we propose a method to adapt to the target domain's texture. First, we diversity the texture of synthetic images using a style transfer algorithm. The various textures of generated images prevent a segmentation model from overfitting to one specific (synthetic) texture. Then, we fine-tune the model with self-training to get direct supervision of the target texture. Our results achieve state-of-the-art performance and we analyze the properties of the model trained on the stylized dataset with extensive experiments.

1. Introduction

Until now, many studies have dealt with semantic segmentation. For supervised semantic segmentation, a large volume of labeled data is required for training. However, the manual annotation for pixel-wise ground truth labels is extremely laborious. For example, it takes 90 min per image to make ground truth label for the Cityscape [5] dataset.

To reduce the cost of annotation, datasets such as GTA5 [20] and SYNTHIA [21] are proposed. Since these datasets are generated by computer graphics, the images and pixel-level annotations are automatically generated. However, due to the domain gap between the synthetic domain and the real domain, a model trained with the synthetic data is hard to generalize to the real data.

Domain adaptation addresses the above issue by reducing the domain gap. One approach is pixel-level adaptation. The pixel-level adaptation uses image translation algorithms like CycleGAN [29] to reduce the gap in visual appearance between two domains. Since the synthetic image is translated into the style of the real domain, a model can learn representation for the real domain more easily.

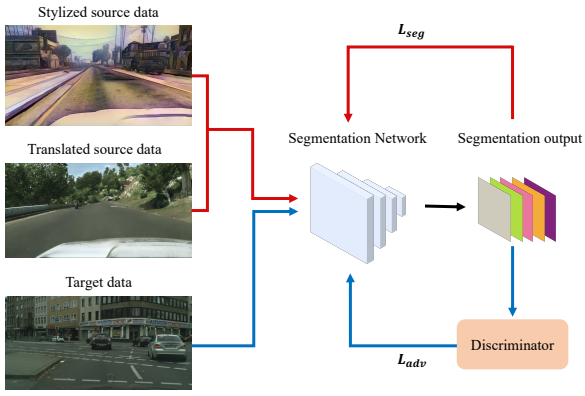

Figure 1: Process of learning texture-invariant representation. We consider both the stylized image and the translated image as the source image. The red line indicates the flow of the source image and the blue line indicates the flow of the target image. By segmentation loss of the stylized source data, the model learns texture-invariant representation. By adversarial loss, the model reduces the distribution gap in feature space.

Although CycleGAN reduces the visual gap between two domains to some extent, overcoming the fundamental difference, the *texture*, is still challenging. In Figure 2, the second column shows translated results by CycleGAN. Although the translated images get the Cityscapes' gray color tone, CycleGAN cannot completely translate the synthetic texture into the real texture. Therefore, the possibility of a model to overfit to the synthetic texture still exists.

To overcome this limitation, we propose a method to adapt to the target domain's texture. First, we generate a texture-diversified source dataset by using a style transfer algorithm. Each source image loses the synthetic texture and gets a random texture. Because of the increased variation of textures, a model trained on the texture-diversified

Figure 2: Texture comparison. Original GTA5 [20] images (first column), generated images by CycleGAN [29] (second column) and by Style-swap [4] (third column).

dataset is guided to learn texture-invariant representation. Then, we fine-tune the model using self-training to get direct supervision of the target texture.

Our method achieves state-of-the-art performance on the GTA5 to Cityscapes benchmark. With extensive experiments, we analyze the properties of the model trained on the stylized dataset and compare the differences between ours and CycleGAN-based methods.

Our contributions are as follows:

1. We design a method to adapt to the target domain's texture for domain adaptation of semantic segmentation, combining pixel-level method and self-training.

2. We achieve state-of-the-art performance on the GTA5 to Cityscapes benchmark.

3. With extensive experiments, we analyze the properties of the model trained on the stylized dataset.

4. We compare our style transfer-based approach and previous CycleGAN-based methods in terms of reducing the domain gap between the synthetic domain and the real domain.

2. Related Work

2.1. Domain adaptation of semantic segmentation

Domain adaptation transfers knowledge between different domains. Assume two datasets that have similar but different distributions. Let the one which has a larger volume and is more easy to collect as the source domain and the other as the target domain. The goal of domain adaptation is transferring knowledge learned from the source domain to the target domain.

Among some settings of domain adaptation, the unsupervised setting is the most popular, which has access to input data and ground truth labels for the source domain but only input data for the target domain. The goal of unsupervised domain adaptation is to use the fully-labeled source domain properly to improve performance on the unlabeled target domain. Since annotating semantic label is one of the most laborious processes, domain adaptation of semantic segmentation gets much attention recently.

Pixel-level adaptation. There exists a visual gap between synthetic and real images, such as texture and lighting. Pixel-level adaptation translates the synthetic source image into the target style using image translation algorithms like CycleGAN [29]. Due to the reduced visual gap, a model more easily encodes the representation for the target domain.

Self-training. Recently, some works adopt self-training (ST) for domain adaptation of semantic segmentation [30, 16]. Generally, ST is applied when labeled training data is scarce. In the unsupervised domain adaptation, because labels of the target domain are absent, it is very attractive to apply ST. [16] suggests a simple method for self-training. At ST stage, [16] generates pseudo labels based on the previous model's confident prediction and fine-tune the model with pseudo labels.

[16] uses both pixel-level adaptation and self-training. In ablation study, the models trained with ST method outperform other models only using the pixel-level method with a large margin. Considering the fundamental difference between the two domains as the *texture*, powerful performance of ST, which gets direct supervision of the target texture, means that previous methods using pixel-level adaptation are not able to encode the target texture sufficiently.

Based on this observation, we propose a method that is optimized for encoding the target domain's texture.

2.2. Style transfer

Starting from texture synthesis [7] and going through [8], many studies have been conducted about style transfer. Based on the observation that style(texture) and content can be separated, modeling feature statistics makes possible to synthesize image with one image's content and another image's texture.

Our purpose is, using various textures as a regularizer preventing a model from overfitting to one specific texture, to make the segmentation model learn texture-invariant representation.

2.3. Texture and shape

According to recent research [9], human recognition is based on shape but the ImageNet [6] pre-trained CNN's criterion is based on texture. To overcome texture-dependency, [9] generates Stylized ImageNet (SIN) using the AdaIN [14] style transfer algorithm. Stylized ImageNet lose natural texture and get the various random texture. Since a model trained on SIN cannot predict results based on the local texture, it is enforced to consider the overall structure of the input. [9] demonstrates with experiments that CNN trained on SIN is more shape-dependent like humans and

(a) Style (b) Content (c) AdaIN (d) PhotoWCT (e) Style-Swap

Figure 3: Results of stylization.

the shape-dependent model is better at classification and detection tasks.

Inspired by this work, we apply this method to domain adaptation of semantic segmentation task, where the texture is fundamental differences between synthetic and real domains.

3. Method

In this section, we present a process for generating texture-diversified datasets and a method to adapt to the target texture. We first diversity the texture of the original source dataset with a style transfer algorithm Style-swap [4] and translate the original source dataset with an image translation algorithm CycleGAN [29]. Then, our model goes through two training stages.

Stage 1: We train a segmentation model with the texture-diversified dataset to learn texture-invariant representation.

Stage 2: Based on the texture-invariant representation, we fine-tune the model to the target domain's texture.

3.1. Stylized GTA5 / SYNTHIA

Prior works [13, 16] use an image translation method CycleGAN [29] to reduce the visual gap between the synthetic and real domains. Although the image translation algorithm makes the source image's overall color tone similar to the real image, it cannot completely translate the synthetic texture into the real one.

To overcome this limitation, we take a more fundamental approach which removes the synthetic texture drastically. Inspired by [9], we generate Stylized GTA5 and Stylized SYNTHIA. Stylized ImageNet [9] is generated by fast AdaIN [14] style transfer algorithm. Although AdaIN is efficient in inference, it distorts the structure of content image considerably with some wave patterns. Unlike the classification task, semantic segmentation task requires accurate pixel-level annotations. Thus, we cannot use AdaIN. The photo-realistic style transfer algorithm [15] is another option, which preserves the precise structure of the original image using a smoothing step after the stylization step. However, due to the smoothing process which is based on the original content image, final results preserve original synthetic texture. Since our purpose is to remove the synthetic texture using a style transfer algorithm, it is not appropriate to apply the photo-realistic algorithm. Our requirements are three-fold. First, enough stylization effect to remove the synthetic texture, while not distorting the structure of the original image too much. Second, due to the large image resolution and the large volume of the synthetic dataset, the stylization process should be time-efficient. Third, to generate diverse stylized results, it should be able to transfer various styles. Considering above conditions, we choose Style-swap [4]. We present stylization results from different methods in Figure 3.

For a style dataset, we used the *Painter by Numbers* dataset which consists of artistic images. Considering the volume of the GTA5 and SYNTHIA dataset, we use the first split, which contains 11,026 images. The stylized datasets have the same number of images with the original datasets, i.e. one-to-one mapping.

As shown in Figure 4, the stylized images drastically lose the synthetic texture and get various random textures. Since each texture is from a different style image, this variety of texture leads a model to encode texture-invariant representation. In other words, the model can learn shape-dependent representation.

3.2. Stage 1

The goal of the first stage is to learn texture-invariant representation using the texture-diversified dataset. We train the segmentation model with both the stylized images by Style-swap [4] and the translated images by CycleGAN [29]. At each iteration, the stylized or translated inputs are alternately forwarded due to the limitation of memory. While learning texture-invariant representation with the stylized images, the translated images guide the model toward the target style.

Along with the texture regularization, we additionally use the output-level adversarial training [23] to further align feature space between the two different domains. The process of Stage 1 is shown in Figure 1.

3.3. Stage 2

The goal of the second stage is, based on learned texture-invariant representation, to fine-tune the segmentation network to the target domain's texture. For this purpose, we adopt a self-training method. Following the process of [16], we generate pseudo labels with the model trained on Stage

Figure 4: Examples of original images and stylized images.

1. Among predictions on the target training images, we set predictions with higher confidence than a threshold(0.9) as pseudo-labels. Then we fine-tune the model with the generated pseudo-labels and translated source images. Now the model is directly supervised by the target domain's texture, the model can learn the representation optimized for the target domain. We apply this process iteratively.

3.4. Training objective

Segmentation model training. Since the ground truth label is only available in the source domain, the segmentation loss is defined as:

$$L_{seg}(I_s) = -\sum_{h,w}\sum_{c=1}^{C} y_s^{h,w,c} \log P_s^{(h,w,c)} \tag{1}$$

And when the target image is given, we calculate the adversarial loss using discriminator.

$$L_{adv}(I_t) = -\sum_{h,w} \log D(P_t^{(h,w,c)}) \tag{2}$$

where I_s and I_t are the input images from the source domain and the target domain. $P_s^{(h,w,c)}$ and $P_t^{(h,w,c)}$ are the final feature of the source and target image. $y_s^{h,w,c}$ is the source domain's ground truth pixel label. C is the number of classes and D is a fully convolutional discriminator.

Therefore, the total loss function for the segmentation network is defined as:

$$L(I_s, I_t) = L_{seg}(I_s) + \lambda_{adv} L_{adv}(I_t) \tag{3}$$

Discriminator Training. The discriminator takes source and target features and classifies whether it is from the source or target domain.

$$\begin{aligned} L_D(P) = -\sum_{h,w}((1-z)\log D(P_s^{(h,w,c)}) \\ + z \log D(P_t^{(h,w,c)})) \end{aligned} \tag{4}$$

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

Self-training. In stage 2, to get direct supervision of the target domain's texture, we calculate the segmentation loss for generated pseudo-labels in target images.

$$L_{ST}(I_t) = -\sum_{h,w} \mathbb{1}_{pseudo} \sum_{c=1}^{C} \hat{y}_t^{h,w,c} \log P_t^{(h,w,c)} \tag{5}$$

where $\mathbb{1}_{pseudo}$ indicates whether each pixel of the target training set is pseudo-label or not.

4. Experiments

Dataset. GTA5 [20] is a dataset which contains 24,966 synthetic images from the video game with 1914×1052 resolution. The semantic labels are compatible with the Cityscapes dataset in 19 classes.

For SYNTHIA [21], we use the SYNTHIA-RAND-CITYSCAPES partition with 9,400 images of 1280×760 resolution. We validate on 13 common classes with the Cityscapes dataset.

Cityscapes [5] is a dataset which contains 5,000 densely annotated images with 2048×1024 resolution. We use 2,975 training images and 500 validation images.

Network architecture. We use the DeepLab-v2 [2] model with ResNet-101 [11] and VGG-16 [22] which are pretrained on ImageNet [6]. For the discriminator, we adopt similar architecture to [19]. The network contains 5 convolution layers with 4×4 kernel size, channel numbers are $\{64,128, 256, 512, 1\}$ and stride of 2.

Training detail. We implement our experiment using the Pytorch library on a single GTX 1080 Ti. To optimize the segmentation model, we use the SGD method. The momentum is set as 0.9. The initial learning rate is 1.0×10^{-4} for Stage 1. Due to the variation of the stylized dataset, a high learning rate makes training unstable. Therefore, we set smaller value than prior works which adopt the same architecture [23, 18, 25, 1, 16]. The same learning rate is used for fine-tuning in Stage 2. For the learning rate schedule, we adopt the polynomial procedure mentioned in [2]. For optimizing discriminator, we use Adam for optimizing

Table 1: Results on GTA5 to Cityscapes.

GTA5 → Cityscapes																					
Base Model	Method	road	side.	buil.	wall	fence	pole	t-light	t-sign	vege.	terr.	sky	pers.	rider	car	truck	bus	train	motor	bike	mIoU
ResNet101	AdaptSegNet[23]	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
	CLAN[18]	87.0	27.1	79.6	27.3	23.3	28.3	35.5	24.2	83.6	27.4	74.2	58.6	28.0	76.2	33.1	36.7	**6.7**	31.9	31.4	43.2
	ADVENT[25]	87.6	21.4	82.0	34.8	26.2	28.5	35.6	23.0	84.5	35.1	76.2	58.6	30.7	**84.8**	34.2	43.4	0.4	28.4	35.2	44.8
	BDL[16]	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
	SIBAN[17]	88.5	35.4	79.5	26.3	24.3	28.5	32.5	18.3	81.2	40.0	76.5	58.1	25.8	82.6	30.3	34.4	3.4	21.6	21.5	42.6
	AdaptPatch[24]	92.3	51.9	82.1	29.2	25.1	24.5	33.8	33.0	82.4	32.8	82.2	58.6	27.2	84.3	33.4	46.3	2.2	29.5	32.3	46.5
	MaxSquare[3]	89.4	43.0	82.1	30.5	21.3	30.3	34.7	24.0	**85.3**	39.4	78.2	**63.0**	22.9	84.6	**36.4**	43.0	5.5	34.7	33.5	46.4
	Ours	**92.9**	**55.0**	**85.3**	34.2	**31.1**	**34.9**	**40.7**	34.0	85.2	40.1	**87.1**	61.0	31.1	82.5	32.3	42.9	0.3	**36.4**	**46.1**	**50.2**
VGG16	AdaptSegNet[23]	87.3	29.8	78.6	21.1	18.2	22.5	21.5	11.0	79.7	29.6	71.3	46.8	6.5	80.1	23.0	26.9	0.0	10.6	0.3	35.0
	CLAN[18]	88.0	30.6	79.2	23.4	20.5	26.1	23.0	14.8	81.6	34.5	72.0	45.8	7.9	80.5	**26.6**	29.9	0.0	10.7	0.0	36.6
	ADVENT[25]	86.8	28.5	78.1	27.6	24.2	20.7	19.3	8.9	78.8	29.3	69.0	47.9	5.9	79.8	25.9	**34.1**	0.0	11.3	0.3	35.6
	BDL[16]	89.2	40.9	81.2	29.1	19.2	14.2	29.0	**19.6**	83.7	35.9	80.7	**54.7**	**23.3**	**82.7**	25.8	28.0	2.3	**25.7**	**19.9**	41.3
	SIBAN[17]	83.4	13.0	77.8	20.4	17.5	24.6	22.8	9.6	81.3	29.6	77.3	42.7	10.9	76.0	22.8	17.9	5.7	14.2	2.0	34.2
	AdaptPatch[24]	87.3	35.7	79.5	32.0	14.5	21.5	24.8	13.7	80.4	32.0	70.5	50.5	16.9	81.0	20.8	28.1	4.1	15.5	4.1	37.5
	DRPC[28]	84.6	31.5	76.3	25.4	17.2	28.2	21.5	13.7	80.7	26.8	74.9	47.5	15.8	77.1	22.2	22.7	1.7	8.9	9.7	36.1
	Ours	**92.5**	**54.5**	**83.9**	**34.5**	**25.5**	**31.0**	**30.4**	18.0	**84.1**	**39.6**	**83.9**	53.6	19.3	81.7	21.1	13.6	**17.7**	12.3	6.5	**42.3**

Table 2: Results on SYNTHIA to Cityscapes.

SYNTHIA → Cityscapes															
Base Model	Method	road	side.	buil.	t-light	t-sign	vege.	sky	pers.	rider	car	bus	motor	bike	mIoU
ResNet101	AdaptSegNet[23]	84.3	42.7	77.5	4.7	7.0	77.9	82.5	54.3	21.0	72.3	32.2	18.9	32.3	46.7
	CLAN[18]	81.3	37.0	80.1	16.1	13.7	78.2	81.5	53.4	21.2	73.0	32.9	22.6	30.7	47.8
	ADVENT[25]	85.6	42.2	79.7	5.4	8.1	80.4	84.1	57.9	23.8	73.3	36.4	14.2	33.0	48.0
	BDL[16]	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**
	SIBAN[17]	82.5	24.0	79.4	**16.5**	12.7	79.2	82.8	**58.3**	18.0	79.3	25.3	17.6	25.9	46.3
	AdaptPatch[24]	82.4	38.0	78.6	3.9	11.1	75.5	**84.6**	53.5	21.6	71.4	32.6	19.3	31.7	46.5
	MaxSquare[3]	82.9	40.7	80.	12.8	**18.2**	**82.5**	82.2	53.1	18.0	79.0	31.4	10.4	35.6	48.2
	Ours	**92.6**	**53.2**	79.2	1.6	7.5	78.6	84.4	52.6	20.0	**82.1**	34.8	14.6	39.4	49.3
VGG16	AdaptSegNet[23]	78.9	29.2	75.5	0.1	4.8	72.6	76.7	43.4	8.8	71.1	16.0	3.6	8.4	37.6
	CLAN[18]	80.4	30.7	74.7	1.4	8.0	77.1	79.0	**46.5**	8.9	73.8	18.2	2.2	9.9	39.3
	ADVENT[25]	67.9	29.4	71.9	0.6	2.6	74.9	74.9	35.4	9.6	67.8	21.4	4.1	15.5	36.6
	SIBAN[17]	70.1	25.7	**80.9**	3.8	7.2	72.3	80.5	43.3	5.0	73.3	16.0	1.7	3.6	37.2
	AdaptPatch[24]	72.6	29.5	77.2	1.4	7.9	73.3	79.0	45.7	14.5	69.4	19.6	7.4	16.5	39.6
	DRPC[28]	77.5	30.7	78.6	**10.6**	**16.1**	75.2	76.5	44.1	**15.8**	69.9	14.7	**8.6**	17.6	41.2
	Ours	**89.8**	**48.6**	78.9	0.0	4.7	**80.6**	**81.7**	36.2	13.0	**74.4**	**22.5**	6.5	**32.8**	**43.8**

method with the learning rate 1.0×10^{-4} and the momentum 0.9 and 0.99. We set λ_{adv} as 0.001. Inputs are resized to 1024×512.

Comparison with state-of-the-art models. As shown in Table 1, our method outperforms all previous state-of-the-art methods on GTA5-to-Cityscapes. BDL [16] iterates the training process six times and outperforms other models with a large margin. Our model surpasses the performance of BDL with only two iterations of the segmentation training as shown in Table 5. These results show that our method (first learn texture-invariant representation, then fine-tune toward target texture) is more effective than a simple self-training method.

For the SYNTHIA to Cityscapes, we compare methods that evaluate performance on 13 classes in Table 2. Our method shows outstanding performance in classes like *road* and *sidewalk*, which occupy large area in input images. Since large-area classes will be more affected by texture, our texture-based method outperforms others in these classes.

Results also report our performance on small classes like *t-light*, *t-sign* and *person* are lower than other methods. Although the texture is a fundamental difference between the synthetic and real domains, it is not the only factor causing the domain gap. The layout gap is also an important factor that we didn't handle in this paper. This layout gap brings discrepancy of shape distribution across domains. In SYNTHIA, *t-light*, *t-sign*, and *person* are depicted much smaller compared to GTA5 and Cityscapes. Since the shape is more decisive factors than texture for small-area classes, our shape-dependent representation, which is fitted to SYNTHIA's shape distribution, is hard to be transferred to Cityscapes' shape distribution.

Also as quantitatively shown in [26], the domain gap be-

tween SYNTHIA and Cityscapes is much larger than the domain gap between GTA5 and Cityscapes, especially for *t-light* and *t-sign*. Other methods use an additional technique like class-ratio prior [25] to reduce the layout gap.

Comparison of class-wise performance. We provide the basis for the above claim through a class-wise ablation study. In Table 3, IoUs are from large (texture-sensitive) and small (texture-insensitive) classes in the Stage 1. Models trained on *Stylized* dataset outperform models trained on *Translated* and *Original* dataset in large-area classes like *road* and *sidewalk*. Among other large-area classes, since *road* and *sidewalk* have similar layout distribution, texture is an especially important factor for these classes.

On the other hand, *Original* outperforms other methods in *t-light* and *t-sign*. [26] shows, when using the synthetic and real data together, performance increases significantly in *t-light* and *t-sign* compared to other classes. This means texture is not a decisive factor for these classes and the sharp original image is more helpful for improving performance in the real domain.

Table 3: Ablation study on large & small classes.

SYNTHIA → Cityscapes					
Base Model	Source Type	road	side.	t-light	t-sign
ResNet101	Stylized	**87.7**	**44.1**	1.0	5.8
	Translated	84.6	40.6	1.3	5.0
	Original [23]	79.2	37.2	**9.9**	**10.5**
VGG16	Stylized	**86.1**	**36.4**	**0.3**	1.7
	Translated	75.6	31.9	0	3.6
	Original [23]	78.9	29.2	0.1	**4.8**

5. Discussion

5.1. Comparison with CycleGAN-based methods

In this section, we compare the differences between ours and CycleGAN-based methods.

First, CyCADA [13] uses CycleGAN to reduce the visual gap between the synthetic and real domains. However, while CycleGAN's generator is trained to generate undistinguishable images from the target domain, CycleGAN is prone to generate inappropriate images.

In Figure 5, for GTA5 to Cityscapes (first row), CycleGAN generates *vegetation*-like artifact on the *sky* to match Cityscapes' distribution. For SYNTHIA to Cityscapes (second row), CycleGAN blurs out *person* to match Cityscapes' color distribution. Despite CycleGAN discriminator's PatchGAN structure, these patterns are easily observed. On the other hand, because Style-swap transfers style based on local patch, Style-swap doesn't show such patterns.

Second, similar to our method, DRPC [28] uses CycleGAN to randomize source images. In Figure 6, we shows

Figure 5: Inappropriate generation of CycleGAN. Original images (first column), generated images by CycleGAN (second column) and Style-swap (third column).

generated images using ImageNet classes used in [28] as auxiliary domains. Figure 7 shows cropped images from Figure 6.

In Figure 7, CycleGAN match auxiliary domain's color distribution while preserving the original image's synthetic texture. On the other hand, Style-swap diversifies texture. This is the most differentiated point of our method compared to DRPC. Since the main difference between the synthetic and real domains is not color but the texture, our texture-based method is more suitable than DRPC for randomization in synthetic to real tasks.

Also, our method is computationally more efficient than DRPC. Since training CycleGAN is a very costly process, DRPC only uses 15 auxiliary domains. On the other hand, since Style-swap does not require additional training for each style, it can handle many styles more easily. Hence our stylized datasets consist of 11,026 styles.

Additionally, DRPC used Pyramid Consistency across Domain (PCD) loss to learn style-invariant feature. Because of this loss, a computation that is linearly proportional to the number of domains is required to simultaneously forward images across domains. Since DRPC used 16 domains, it requires at least 16 times more memory and computing power.

Though DRPC used 16 domains, it might be required to consider more domains for more style-invariant representation, which demands impractical computation especially when the input's resolution is large like GTA5 (1914x1052) and SYNTHIA (1280x760). On the other hand, our method requires a fixed amount of computation regardless of the number of styles.

5.2. Ablation study

We conduct an ablation study on Stage 1 in Table 4. We divide the table into two sections according to the usage of adversarial loss.

In first section, *Original source only* means training the segmentation network only with the original GTA5 images. *Stylized source only* and *Translated source only* use gener-

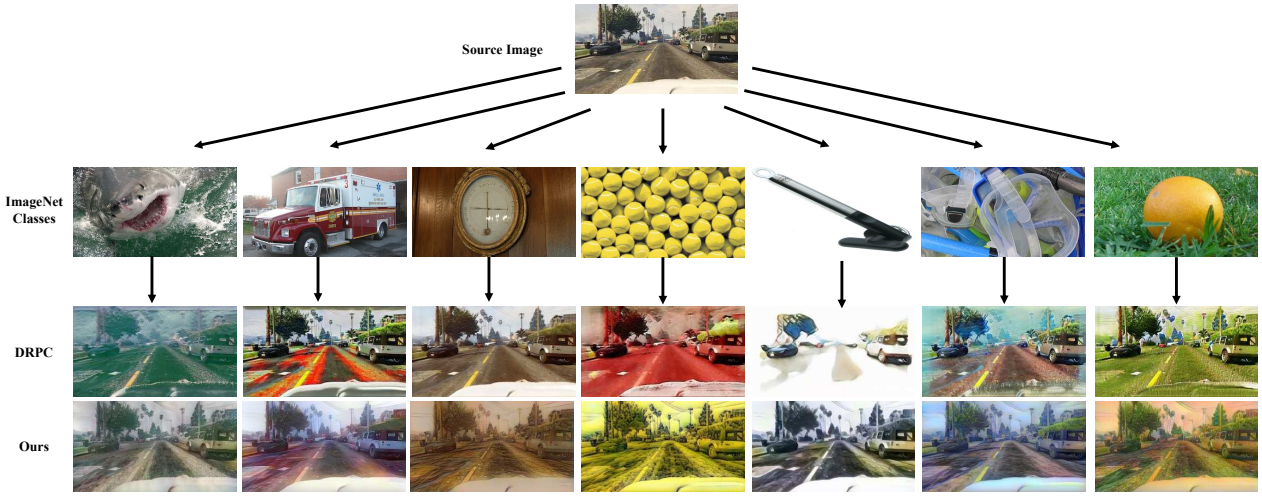

Figure 6: Stylization comparison with DRPC.

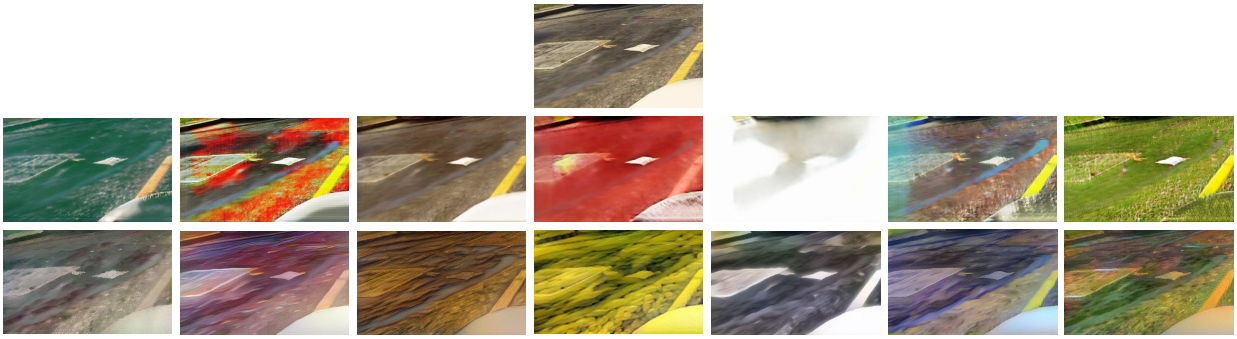

Figure 7: Texture comparison with DRPC. Cropped images from corresponding images from Figure 6.

ated dataset by Style-swap [4] and CycleGAN [29] respectively. Results show model trained on only stylized source dataset outperforms other pixel-level [27, 10] method.

The second section shows the results with the output-level adversarial training [23]. Overall performances are improved compared to the first section. Results show using both types (*Stylized source* and *Translated source*) by forwarding images alternately is better than only using *Stylized source* images. This is because, while learning texture-invariant representation through the stylized images, the translated images guide the model to the target style. Following the results, we choose to use both *Stylized source* *Translated source* images with the output-level adversarial loss for the training segmentation network in Stage 1.

We also conduct the ablation study for Stage 2 in Table 5. The results show in the third iteration of Stage 2 the segmentation model converged. Therefore we take three iterations for all results in Table 1, 2.

Table 4: Ablation study on Stage 1.

GTA5 → Cityscapes	
method	mIoU
Original source only	36.6
DCAN [27]	38.5
Translated source only	41.0
DLOW [10]	42.3
Stylized source only	**42.5**
Original source + Adv loss [23]	41.4
Translated source + Adv loss [16]	42.7
Stylized source + Adv loss	43.2
Stylized/translated source + Adv loss	**44.6**

5.3. Robustness test

To verify the texture-invariance of a model trained on the stylized dataset, we test the model on perturbated validation

Table 5: Ablation study on Stage 2. In Stage 2-X, X means the number of iteration of self training.

GTA5 → Cityscapes	
method	mIoU
Stage 1	44.6
Stage 2-1	48.6
Stage 2-2	50.2
Stage 2-3	50.2

sets distorted by various noises. If the model is texture-invariant, it will be more robust to noises than other texture-dependent models. We generate noisy Cityscapes validation sets with noises that do not distort the shape of the original image's object. Following the method of [12], we add Gaussian, Impulse, Shot and Speckle noise to the validation set.

Results in Table 6 and Figure 8 show that our model is much more robust to various noises than AdaptSegNet [23] which is trained on original synthetic images.

Table 6: Results on original and noisy validation set.

Method	AdaptSegNet[23]	Stylized source only
Original	42.4	42.5
Gaussian	22.2	**35.1**
Impulse	20.9	**32.6**
Shot	24.9	**38.2**
Speckle	32.5	**41.1**

(a) Original image (b) Image with Gaussian noise

(c) AdaptSegNet (d) Ours (e) Ground Truth

Figure 8: Results on the validation image with Gaussian noise.

5.4. Qualitative results

To qualitatively demonstrate the texture-invariance of our model, we present segmentation results on images with various texture from the stylized source dataset in Figure 9. Results show our model is robust to texture variation.

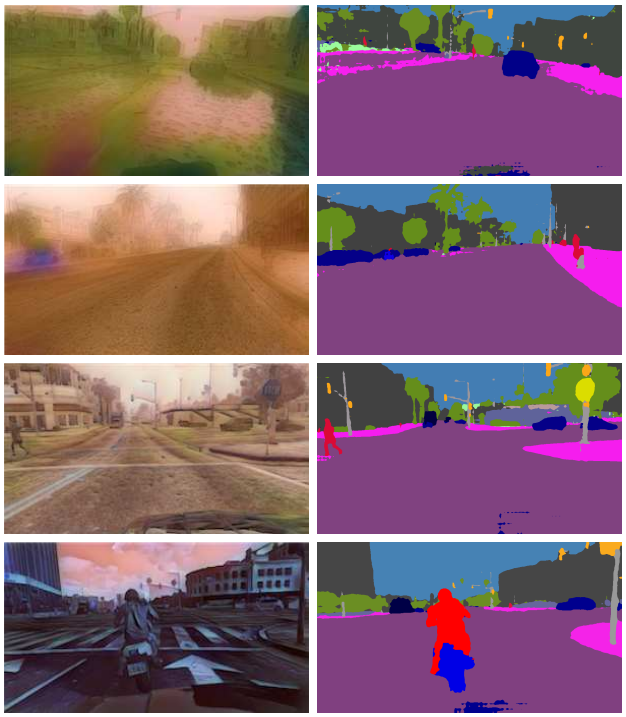

Figure 9: Results on images with various texture. Images from the Stylized GTA5 (left column) and segmentation results (right column).

6. Conclusion

In this paper, we present a method to adapt to the target texture. Using a style transfer algorithm, we generate the Stylized GTA5/SYNTHIA. The various texture of the stylized datasets works as a regularizer to make the segmentation model learn texture-invariant representation. We show the texture-invariance of our model qualitatively on images with various texture and quantitatively on noisy validation sets. Based on the texture-invariant representation, we use self-training to get direct supervision of the target texture. Experimental results show the effectiveness of our approach, which achieves new state-of-the-art performance in the GTA5 to Cityscapes benchmark. Besides, we analyze the influence of texture across different classes. Also, we compare our style transfer-based method and CycleGAN-based methods in terms of reducing the texture gap between the synthetic and real domains.

7. Acknowledgement

This work was supported by the National Research Foundation of Korea grant funded by Korean government (No. NRF-2019R1A2C2003760).

References

[1] Wei-Lun Chang, Hui-Po Wang, Wen-Hsiao Peng, and Wei-Chen Chiu. All about structure: Adapting structural information across domains for boosting semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1900–1909, 2019. 4

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

[3] Minghao Chen, Hongyang Xue, and Deng Cai. Domain adaptation for semantic segmentation with maximum squares loss. *arXiv preprint arXiv:1909.13589*, 2019. 5

[4] Tian Qi Chen and Mark Schmidt. Fast patch-based style transfer of arbitrary style. *arXiv preprint arXiv:1612.04337*, 2016. 2, 3, 7

[5] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 1, 4

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2, 4

[7] Leon Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. In *Advances in neural information processing systems*, pages 262–270, 2015. 2

[8] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016. 2

[9] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018. 2, 3

[10] Rui Gong, Wen Li, Yuhua Chen, and Luc Van Gool. Dlow: Domain flow for adaptation and generalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2477–2486, 2019. 7

[11] 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. 4

[12] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019. 8

[13] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. *arXiv preprint arXiv:1711.03213*, 2017. 3, 6

[14] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017. 2, 3

[15] Yijun Li, Ming-Yu Liu, Xueting Li, Ming-Hsuan Yang, and Jan Kautz. A closed-form solution to photorealistic image stylization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 453–468, 2018. 3

[16] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6936–6945, 2019. 2, 3, 4, 5, 7

[17] Yawei Luo, Ping Liu, Tao Guan, Junqing Yu, and Yi Yang. Significance-aware information bottleneck for domain adaptive semantic segmentation. *arXiv preprint arXiv:1904.00876*, 2019. 5

[18] 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. 4, 5

[19] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 4

[20] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *European conference on computer vision*, pages 102–118. Springer, 2016. 1, 2, 4

[21] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3234–3243, 2016. 1, 4

[22] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 4

[23] 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. 3, 4, 5, 6, 7, 8

[24] Yi-Hsuan Tsai, Kihyuk Sohn, Samuel Schulter, and Manmohan Chandraker. Domain adaptation for structured output via discriminative representations. *arXiv preprint arXiv:1901.05427*, 2019. 5

[25] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2517–2526, 2019. 4, 5, 6

[26] Magnus Wrenninge and Jonas Unger. Synscapes: A photorealistic synthetic dataset for street scene parsing. *arXiv preprint arXiv:1810.08705*, 2018. 5, 6

[27] Zuxuan Wu, Xintong Han, Yen-Liang Lin, Mustafa Gokhan Uzunbas, Tom Goldstein, Ser Nam Lim, and Larry S

Davis. Dcan: Dual channel-wise alignment networks for unsupervised scene adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 518–534, 2018. 7

[28] Xiangyu Yue, Yang Zhang, Sicheng Zhao, Alberto Sangiovanni-Vincentelli, Kurt Keutzer, and Boqing Gong. Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. *arXiv preprint arXiv:1909.00889*, 2019. 5, 6

[29] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 1, 2, 3, 7

[30] Yang Zou, Zhiding Yu, BVK Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 289–305, 2018. 2