SYNTHETIC-TO-REAL GENERALIZATION FOR SEMANTIC SEGMENTATION

*Xinhui Li, Liang Li, Xiaojie Guo**

College of Intelligence and Computing, Tianjin University, Tianjin, China
{lixinhui, liangli}@tju.edu.cn, xj.max.guo@gmail.com

ABSTRACT

The discrepancy between synthetic and real data is crucial to the performance of domain generalization for semantic segmentation. Since real data is not always accessible, a popular line of approaches is to enhance the diversity of synthetic data via either complex adversarial generation or unstable stylization. However, the internal structure of the synthetic image is often neglected. To largely explore useful information in synthetic data, we observe that, although objects of the same category have different texture patterns between domains, their shapes are quite similar. Based on this observation, we argue that focusing on structural information and alleviating texture dependence are effective ways to improve generalization capability. In this work, we propose an end-to-end network, which explicitly constrains the network to learn shapes and spatial knowledge, and implicitly relieves the texture reliance of the network. Extensive experiments verify the effectiveness of our proposed method and demonstrate its clear advantages over other competitors.

***Index Terms*—** Domain generalization; Synthetic-to-real; Semantic segmentation

1. INTRODUCTION

Semantic segmentation has been one of the most typical and essential tasks in computer vision and multimedia, where large-scale annotated real images are usually required. Since pixel-wise labeling is extremely time-consuming and laborious, domain adaptation [1] and domain generalization [2, 3] based on synthetic data training have drawn growing attention. Different from domain adaptation that has both synthetic images and real images available in the training phase, domain generalization only utilizes synthetic images for training and then test on unseen real images. Thus, domain generalization is more challenging than adaption. However, due to the limitation of rendering quality on synthetic data, the texture discrepancy between the synthetic and real data would induce the degradation of network generalization. To enhance the generalization capability, recent approaches in stylized transformation [4, 5], adversarial generation [6], and representation learning [7] are adapted to mitigate the gap between domains. Despite the successes of these methods, the preprocessing of synthetic data seems tedious and the content of generated texture-diversified synthetic images cannot be guaranteed.

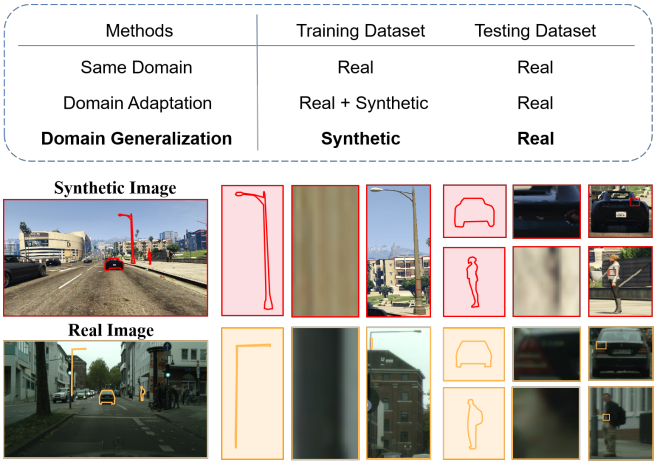


Fig. 1: The settings of different methods on real and synthetic datasets during training and testing. A visual comparison of discrepancy in the shape and texture of the same objects between the synthetic image and real image.

To address the aforementioned limitations, we consider taking advantage of the internal structure in synthetic images themselves and effectively eliminating/reducing the need for complex pre-processing steps. As can be viewed in Fig.1, the shapes of objects such as persons, vehicles, and traffic lights in the synthetic domain are similar to those in the real world, while the textures of the objects are often quite different. In other words, there is a slight difference in terms of shape and spatial information of objects between the synthetic image and the real image, whereas a great texture discrepancy appears. Motivated by this observation, we propose an end-to-end domain generalization network without the requirement of pre-processing, which consists of two modules, including the cross-layer module Spatial Structure Intensifier (SSI) and the adjacent layer module Texture Structure Generalizer (TSG). Due to similar shapes of objects, SSI aims to enhance the generalization ability of the network by learning the (almost) invariant shape and spatial feature representation. Meanwhile, because of the strong difference in ob-

*corresponding author.

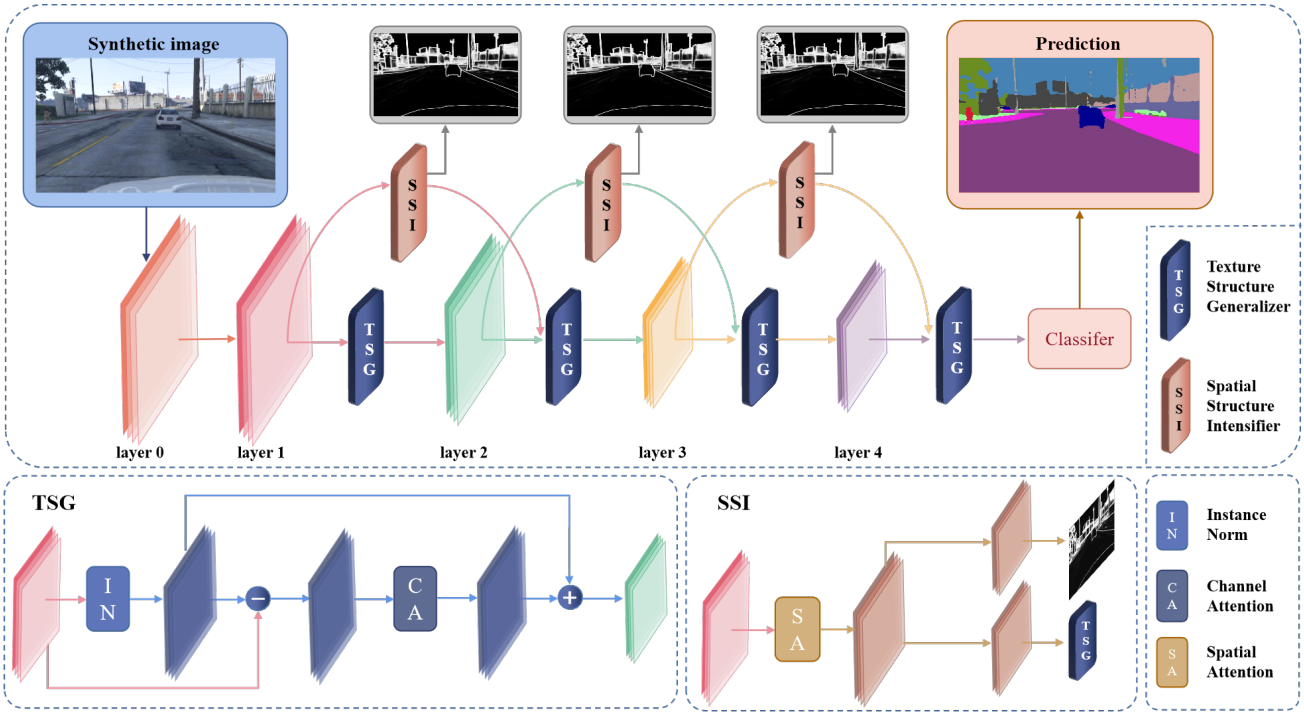


Fig. 2: The overall architecture of the proposed framework is shown in the top dotted box. The bottom two dotted boxes illustrate the internal structure of Texture Structure Generalizer (TSG) module and Spatial Structure Intensifier (SSI) module. The lines in different colors represent the feature flow direction of feature layers.

ject texture, TSG is designed to alleviate the reliance of network classification on specific textures. Both SSI and TSG are interspersed in the network architecture, one by enhancing domain-invariant learning, the other by reducing the impact of domain variants, ultimately achieving the purpose of improving network domain generalization ability. The main contributions of this paper can be summarized as follows:

- This paper proposes a network for synthetic-to-real generalization, the goal of which is to improve generalization ability through seeking internal structures and relieving texture reliance.
- We customize two efficient and flexible modules, *i.e.* Spatial Structure Intensifier (SSI) and Texture Structure Generalizer (TSG), to help achieve the generalization by enhancing shape learning and normalizing texture feature representation, respectively.
- Experiments are conducted to reveal the superiority of our approach compared with methods without pre-processing. Also, we achieve competitive performance with those using pre-processing.

2. RELATED WORK

This section will briefly review representative works in domain adaptation and domain generalization, which are mainly used to address the problem of semantic segmentation when real annotated images are invalid.

Domain Adaptation. Domain adaptation can be performed using unlabeled data from the real domain and labeled data from the synthetic domain. The key to domain adaptation is to reduce the gap between the distributions of data in real and synthetic domains. The common way is directly using maximum mean discrepancies (MMD) [8, 9] to minimize the difference between the distributions of two domains. However, such metrics are limited as minimized MMD cannot guarantee the two domains are well-aligned. Image-to-image translation and style transfer [4, 5] are two representative manners to reduce the discrepancy at the input level of two domains by converting the style of the synthetic image into the real image. Meanwhile, adversarial learning methods [10, 11], which usually rely on the discriminator mechanism to maximize the confusion between domains, can implicitly align the cross-domain features [12] and improve the accuracy of the classifier. Unfortunately, the real data is not always accessible during training and the distribution discrepancy cannot be estimated from a single domain. Therefore, these techniques can hardly be applied to domain generalization unmodified.

Domain Generalization. Compared with domain adaptation, domain generalization purely utilize the synthetic images [13] in training and yet aims to generalize well on the real domain. To improve the generalization performance of networks, a number of appealing methods have been pro-

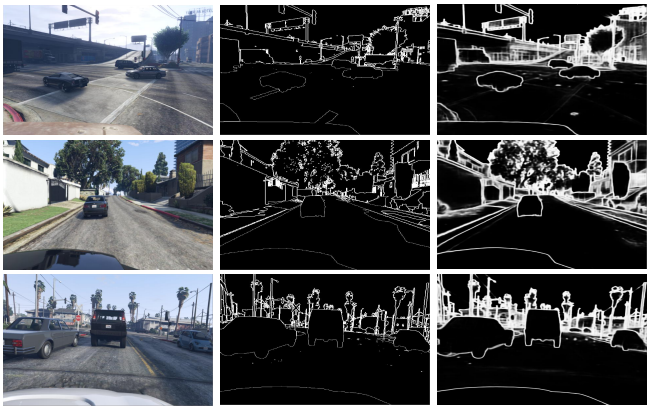
posed. Some methods leverage multiple source domains or image randomization [14, 3] to boost the diversity of inputs explicitly. In addition, several works expand source data by generating new stylized synthetic images [15] through adversarial learning and style transfer [5]. However, it is inflexible to perform complex adversarial generation or unstable stylized transformation on the synthetic image. Another technical line for solving this problem is from the perspective of latent feature representation [2], such as Instance Normalization and Batch Normalization. Inspired by these methods, we consider enhancing the classification ability of the network based on spatial information while using texture generalization, and at the same time do not perform any complex pre-processing on data during training.

3. METHODOLOGY

The focus of this work is on how to solve the domain generalization problem: a model is trained on the synthetic domain, expecting to generalize well on many unseen real-world domains. Due to the discrepancy between domains, as previously analyzed, our method aims to increase the domain generalization ability by enhancing spatial feature learning and generalizing special textures on the synthetic data without pre-processing. Towards this purpose, two efficient modules are designed, namely Spatial Structure Intensifier (SSI) module and Texture Structure Generalizer (TSG) module. In what follows, we will first describe the overall architecture of our method, and then detail the designed TSG and SSI.

3.1. Overview of the proposed framework

The overall framework is simple as depicted in Fig.2. Given a synthetic image as input, here we do not execute any preprocessing like stylization and randomization. Starting from the architecture of feature extraction network ResNet-50, we insert two types of modules among the feature layers: (1) adjacent layer module TSG and (2) cross-layer module SSI. More specifically, we add texture feature generalization TSG after each feature layer, so that texture can be normalized layer by layer. This dense connection can effectively alleviate texture dependence between each feature extraction layer, thus improving the generalization ability of the network. Furthermore, cross-layer spatial structure intensifier SSI is designed to fuse the previous features into the interval feature layer, which is explicitly supervised by edge information. Importantly, the features after SSI are first added to the features of the layer and then enter into the TSG module to avoid introducing texture features that have not been normalized previously. Finally, the feature of the last TSG module is fed forward into the semantic segmentation module to generate the final predictions.



Synthetic Image GT Edge Predicted Edge

Fig. 3: Predicted edge maps of synthetic images, which are output through a branch of the SSI module.

3.2. Texture Structure Generalizer (TSG)

Texture Structure Generalizer (TSG) is proposed to generalize the texture on the feature level. Pixel-level semantic segmentation relies heavily on the texture of the object, therefore, the network trained with the synthetic data cannot perform well in the unseen real data due to the texture discrepancy. To address this problem, texture generalization is an option via alleviating the reliance of networks on texture. Recent studies [16, 5] have found that for complex appearances, such as style or texture, this information can be encoded in the mean and variance of hidden feature layers. Hence, the Instance Normalization (IN) [17] layer shows the potential to effectively eliminate the apparent discrepancy. However, the IN operation is always accompanied by some content losses. Based on these considerations, we first normalize the style of features through IN to generalize the texture and then add the channel attention mechanism to retain useful information.

As shown in Fig.2, we subtract the original feature from the feature after IN. Next, the channel attention mechanism is implemented to extract useful information from the subtracted feature. Finally, the extracted information and the feature after IN are added and sent into the next feature layer, which ensures the integrity of the content largely. The output feature of TSG can be obtained by:

$$F^{\text{output}} = CA\left(F^{\text{input}} - IN\left(F^{\text{input}}\right)\right) + IN\left(F^{\text{input}}\right), \quad (1)$$

where CA denotes the channel attention mechanism [18], while IN stands for the Instance Normalization. In our network, four TSG modules, respectively attached after feature layers 1, 2, 3, and 4, are utilized. Therefore, the feature of each layer can be efficiently normalized by TSG modules and the model can achieve better texture generalization.

3.3. Spatial Structure Intensifier (SSI)

Spatial Structure Intensifier (SSI) is built to boost the ability of network classification through spatial information. Al-

though rendering techniques are still limited, current schemes have made the appearance of the synthetic object look almost identical to the real. For example, as can be viewed in Fig.1, the shapes of vehicles, light poles, persons, and other objects are similar in the synthetic and real images. To this end, we design the SSI module, which utilizes the spatial attention mechanism and generates the predictive edge map to pay attention to spatial and shape features. We adopt three SSI modules, respectively attached after feature layers 1, 2, and 3. Employing interval connection between feature layers, low-level spatial features can incorporate into high-level ones. In this way, the network can retain more high-resolution shape features, thus improving the image segmentation accuracy.

For more details, the input feature is firstly enhanced by the spatial attention mechanism [18] and then divided into two branches. One branch adjusts the number of channels through the 1×1 convolutional layer and then inputs to the next TSG. The other branch outputs the predicted edge map through a module containing three convolution layers and calculates the boundary loss with the ground truth generated by semantic labels. In addition, an extra convolution layer is used to further fuse the three SSI output edge images into one. This supervised training can explicitly assist the SSI module in learning the spatial information of images. Figure 2 exhibits the structure of the edge extraction module and Figure 3 shows the edge prediction results.

3.4. Objective Function

Our model is trained with supervised semantic segmentation loss and boundary loss on the synthetic domain. We define X as the input image, Y as its respective boundary ground truth, and $\widehat{Y}$ as a set of predicted edge prediction. $\widehat{Y} = [\hat{y}_1, \hat{y}_2, \hat{y}_3, \hat{y}_f]$, where $\hat{y}_i$ has the same size as Y. $\hat{y}_1, \hat{y}_2, \hat{y}_3$ are the edge predictions of three SSI modules respectively, and $\hat{y}_f$ is the result of the fused edge map. As the model is deep supervised, class-balanced cross-entropy loss is used as the boundary loss. A single edge prediction loss function can simply imposed as the following:

$$\begin{aligned}\ell^n_{\text{boundary}}\ (W, w^n) = &-\beta \sum_{j\in Y_+} \log \sigma\left(y_j = 1 \mid X; W, w^n\right) \\ &-(1-\beta) \sum_{j\in Y_-} \log \sigma\left(y_j = 0 \mid X; W, w^n\right).\end{aligned} \tag{2}$$

Then, the boundary loss $\mathcal{L}_{\text{boundary}}$ is the sum of each ℓ_{boundary}, which can be written as:

$$\mathcal{L}_{\text{boundary}} = \sum_{n=1}^{4} \ell^n_{\text{boundary}}\ (W, w^n), \tag{3}$$

where $\sigma(\cdot)$ is the sigmoid function. W is the collection of network parameters and w is the n corresponding parameter. $\beta = |Y_-| \,/\, (|Y_-| + |Y_+|)$, and $(1-\beta) = |Y_+| \,/\, (|Y_-| + |Y_+|)$. $|Y_-|$ and $|Y_+|$ denote the edge and non-edge in the ground truth. In addition, the semantic segmentation loss given by standard cross-entropy loss is defined as:

$$\mathcal{L}_{\text{seg}} = -\sum_{h,w}\sum_{c=1}^{C} y_s^{(h,w,c)} \log p_s^{(h,w,c)}, \tag{4}$$

where $p_s^{(h,w,c)}$ is the predicted semantic segmentation result. $y_s^{(h,w,c)}$ is the semantic ground truth label. C is the number of classes.

Combining the above two terms yields our final objective function as follows:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{seg}} + \lambda \cdot \mathcal{L}_{\text{boundary}}, \tag{5}$$

where λ is the hyper-parameter used to balance the importance of semantic segmentation loss and boundary loss. In this paper, λ is empirically set to 1.5.

Table 1: Quantitative results of domain generalization from GTA5 to Cityscapes. We measure the mIoU performance of the 19 classes in the validation set of Cityscapes. The backbone of these methods all uses ResNet-50. BS denotes the batch size.

Methods	Pre-processing	BS	mIoU%	mIoU↑%
baseline	-	16	22.17	7.47↑
IBN-Net [2]			29.64	
baseline	✓	32	32.45	4.97↑
DRPC [19]			37.42	
baseline	✓	6	25.88	**9.39**↑
CSG [20]			35.27	
baseline	-	2	28.95	7.63↑
RobustNet [21]			36.58	
baseline	✓	2	31.70	6.90↑
Peng et al. [3]			38.60	
baseline	-	1	26.21	**8.03**↑
Ours			34.24	

4. EXPERIMENTS

In this section, we present the experimental results and compare them with other domain generalization methods on the semantic segmentation task. We also analyze the performance of the proposed modules through the ablation studies.

4.1. Experiment setup

Synthetic Dataset: In the experiments of domain generalization on semantic segmentation, we adopt the dataset GTA5 [13] for training. GTA5 is a large-scale synthetic dataset containing 24966 urban scene images, which are rendered by Grand Theft Auto V game engine and automatically per-pixel annotated into different semantic categories. The original GTA5 dataset provides pixel-level semantic annotations

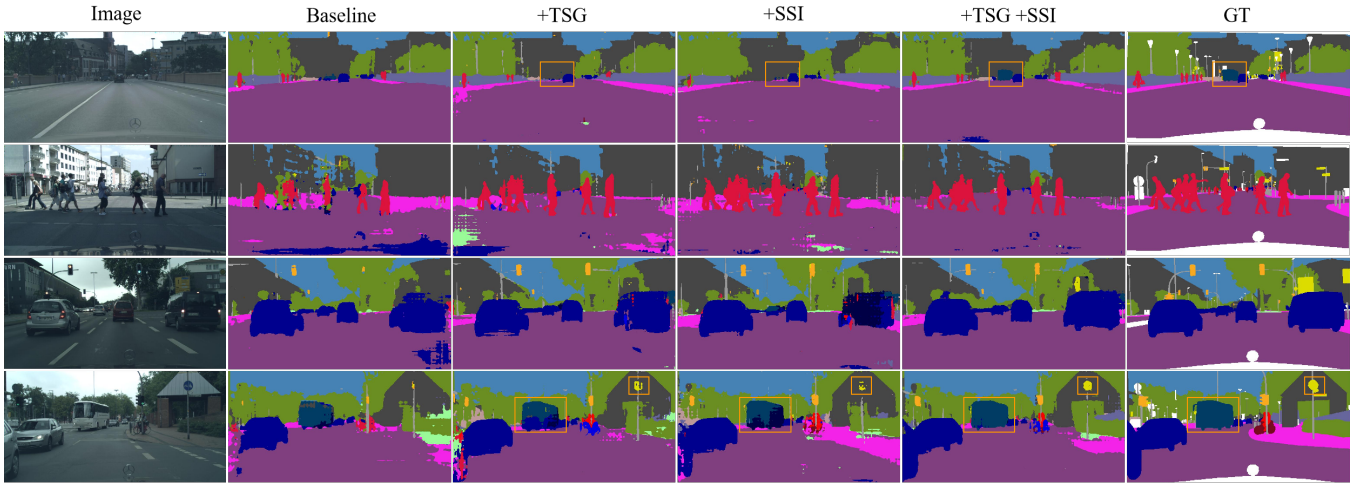


Fig. 4: Visual results of the semantic segmentation on the real dataset Cityscapes. Both TSG and SSI modules effectively improve the performance of the framework.

Table 2: Performance contribution of our designed modules.

Network	TSG	SSI	mIoU %	mIoU↑ %
ResNet-50			26.21	-
	✓		31.72	5.51↑
		✓	32.23	6.02↑
	✓	✓	34.24	8.03↑

of 33 classes and shares the same set of 19 semantic classes with the real-world dataset.

Real-world Dataset: To evaluate the generalization capability, we choose the validation split of real-world dataset Cityscapes [22] which is unseen during training. Cityscapes is a semantic segmentation dataset collected in street scenarios, which contains a training set with 2975 images and a validation set with 500 images. The images are annotated into 19 classes. We only utilize the validation set to test the performance of our model for comparison with other approaches.

Implementation Details: Our Network is implemented in PyTorch and runs on a single NVIDIA RTX2080Ti GPU. For the training set in GTA5, we resize the input image to 640×640 with random cropping and flipping. For the testing set in Cityscapes, the images are simply resized to 1024×512. Model is trained for 20 epochs without using pre-trained parameters. In the training period, we choose Stochastic Gradient Descent (SGD) optimizer with a learning rate of 2.5e-4, a momentum of 0.9, a weight decay of 0.0005, and a batch size of 1. For comparison with prior work on domain generalization, we utilize ResNet-50 as the backbone. We choose PASCAL VOC Intersection over Union (IoU) as the evaluation metric for testing. mIoU is the mean value of IoUs across all categories.

4.2. Comparison with State-of-the-Art

We compare our method with recent state-of-the-art domain generalization methods, which include IBN-Net [2], DRPC [19], CSG [20], RobustNet [21], and Peng et al. [3]. All methods are trained using the synthetic dataset GTA5 and then test on the validation set of the real-world dataset Cityscapes. Some of these methods involve complex pre-processing, such as adversarial generation and image randomization, while others do not utilize pre-processing. In order to verify the effectiveness of our method to the greatest extent, we set the batch size to 1 without pre-trained parameters during the training. Table 1 shows the quantitative results of semantic segmentation accuracy mIoU from GTA5 to Cityscapes. Baseline is trained by GTA5 using network ResNet-50 only. Our proposed method improves by 8.03% on mIoU compared to the benchmark network. We can observe our method outperforms the other methods when there is no pre-processing used, like IBN-Net and RobustNet. Moreover, our method also achieves a competitive result compared to the state-of-the-art method with pre-processing.

Table 3: The effect of hyper-parameter λ, which is designed to trade off the quality of segmentation loss and boundary loss.

Network	mIoU %	mIoU↑ %
baseline	26.21	-
λ =1.00	32.68	6.47 ↑
λ =1.25	33.43	7.22 ↑
λ =1.50	34.24	8.03 ↑
λ =1.75	33.34	7.13 ↑
λ =2.00	32.59	6.38 ↑

4.3. Ablation Studies

To further assess the the significance of each component in our proposed approach, we conduct different types of ablation studies. We first verify the the effectiveness of designed two modules: Texture Structure Generalizer and Spatial Struc-

ture Intensifier. Table 2 shows the mIoU improvement by adding our designed modules. When the TSG or SSI module is adopted alone, mIoU is improved from 26.21 to 31.72 and 26.21 to 32.23, respectively. When two modules are added simultaneously, the mIoU achieves 34.24. These results show that taking two proposed modules is beneficial to generalization performance. Meanwhile, we also visualize the segmentation results, including the results of baseline, single module network, and the entire network. As shown in Fig.4, we use the orange box to mark the difference of segmentation results after adding TSG and SSI modules. Better results can be achieved when using two modules at the same time, such as the segmentation of traffic signs, vehicles, and pedestrians. Especially in the fourth line of Fig.4, the segmentation of the traffic sign and the bus combined with TSG and SSI is better than that of a single module. In addition, there is a significant hyper-parameter λ in equation 5 to trade off the quality of boundary loss with segmentation loss. To evaluate the influence of λ on our model, we set λ from 1.00 to 2.00. As shown in Table 3, we can observe that when $\lambda = 1.5$, our method achieves the best performance.

5. CONCLUSION

In this paper, we have presented a simple, efficient, and no-preprocessing network of domain generalization for semantic segmentation. The internal structure of the synthetic image has been exploited to enhance the generalization capability of the network. To relieve the reliance of network classification on texture, we proposed a Texture Structure Generalizer (TSG) module. In addition, Spatial Structure Intensifier (SSI) is designed to enhance the segmentation capability of the network through learning spatial knowledge. These two proposed modules can be flexibly inserted among feature extraction layers in the network by adjacent layer and cross-layer. Experimental results on the Cityscapes dataset have confirmed the superiority of our method over prior methods.

Acknowledgement

This work was supported by the National Natural Science Foundation of China under Grant no. 62072327, and TSTC under Grant no. 20JCQNJC01510.

6. REFERENCES

[1] Y. Li, L. Yuan, and N. Vasconcelos, "Bidirectional learning for domain adaptation of semantic segmentation," in *CVPR*, 2019.

[2] X. Pan, P. Luo, J. Shi, and X. Tang, "Two at once: Enhancing learning and generalization capacities via ibn-net," in *ECCV*, 2018.

[3] D. Peng, Y. Lei, L. Liu, P. Zhang, and J. Liu, "Global and local texture randomization for synthetic-to-real semantic segmentation," *IEEE TIP*, vol. 30, pp. 6594–6608, 2021.

[4] J. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *ICCV*, 2017.

[5] X. Huang and S. J. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *ICCV*, 2017.

[6] K. Zhou, Y. Yang, T. M. Hospedales, and T. Xiang, "Deep domain-adversarial image generation for domain generalisation," in *AAAI*, 2020.

[7] Y. Li, M. Gong, X. Tian, T. Liu, and D. Tao, "Domain generalization via conditional invariant representations," in *AAAI*, 2018.

[8] B. Geng, D. Tao, and C. Xu, "Daml: Domain adaptation metric learning," *IEEE TIP*, vol. 20, pp. 2980–2989, 2011.

[9] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," in *ICML*, 2015.

[10] T. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez, "Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation," in *CVPR*, 2019.

[11] J. Yang, R. Xu, R. Li, X. Qi, X. Shen, G. Li, and L. Lin, "An adversarial perturbation oriented domain adaptation approach for semantic segmentation," in *AAAI*, 2020.

[12] M. Kim, S. Joung, S. Kim, J. Park, I. Kim, and K. Sohn, "Cross-domain grouping and alignment for domain adaptive semantic segmentation," in *AAAI*, 2021.

[13] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *ECCV*, 2016.

[14] A. Prakash, S. Boochoon, M. Brophy, D. Acuna, E. Cameracci, G. State, O. Shapira, and S. Birchfield, "Structured domain randomization: Bridging the reality gap by context-aware synthetic data," in *ICRA*, 2019.

[15] M. Kim and H. Byun, "Learning texture invariant representation for domain adaptation of semantic segmentation," in *CVPR*, 2020.

[16] V. Dumoulin, J. Shlens, and M. Kudlur, "A learned representation for artistic style," in *ICLR*, 2017.

[17] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky, "Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis," in *CVPR*, 2017.

[18] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," *IEEE TPAMI*, vol. 42, pp. 2011–2023, 2020.

[19] X. Yue, Y. Zhang, S. Zhao, A. L. Sangiovanni-Vincentelli, K. Keutzer, and B. Gong, "Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data," in *ICCV*, 2019.

[20] W. Chen, Z. Yu, S. De Mello, S. Liu, J. Manuel Álvarez, Z. Wang, and A. Anandkumar, "Contrastive syn-to-real generalization," in *ICLR*, 2021.

[21] S. Choi, S. Jung, H. Yun, J. Taery Kim, S. Kim, and J. Choo, "Robustnet: Improving domain generalization in urban-scene segmentation via instance selective whitening," in *CVPR*, 2021.

[22] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *CVPR*, 2016.