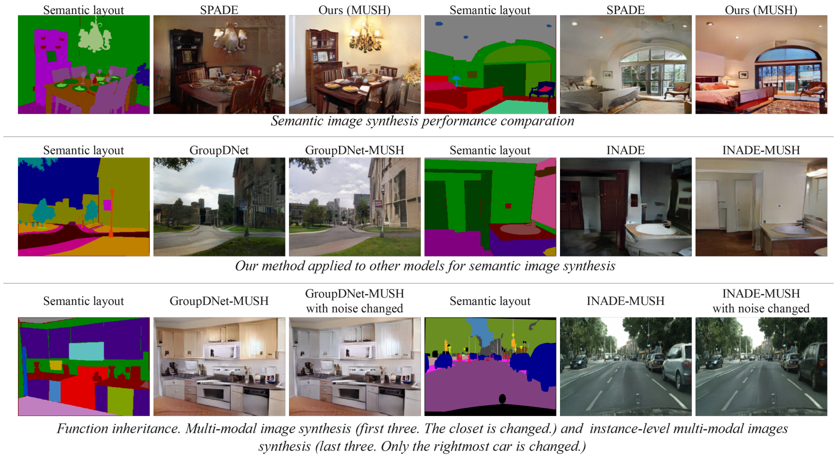
MUSH: Multi-Scale Hierarchical Feature Extraction for Semantic Image Synthesis

Zicong Wang[1,2], Qiang Ren[1,2], Junli Wang[1,2], Chungang Yan[1,2], and Changjun Jiang[1,2]

[1] Key Laboratory of Embedded System and Service Computing (Tongji University),
Ministry of Education, Shanghai 201804, China
[2] National (Province-Ministry Joint) Collaborative Innovation Center for Financial
Network Security, Tongji University, Shanghai 201804, China.
{wangzicong, rqfzpy, junliwang, yanchungang, cjjiang}@tongji.edu.cn



Semantic image synthesis performance comparation

Our method applied to other models for semantic image synthesis

Function inheritance. Multi-modal image synthesis (first three. The closet is changed.) and instance-level multi-modal images synthesis (last three. Only the rightmost car is changed.)

Fig. 1. Our method improves image synthesis performance by utilizing multi-scale information. It can be applied to many models and get better results.

Abstract. Semantic image synthesis aims to translate semantic label masks to photo-realistic images. Previous methods have limitations that extract semantic features with limited convolutional kernels and ignores some crucial information, such as relative positions of pixels. To address these issues, we propose MUSH, a novel semantic image synthesis model that utilizes multi-scale information. In the generative network stage, a multi-scale hierarchical architecture is proposed for feature extraction and merged successfully with guided sampling operation to enhance semantic image synthesis. Meanwhile, in the discriminative network stage, the model contains two different modules for feature extraction of semantic masks and real images, respectively, which helps use semantic masks information more effectively. Furthermore, our proposed model achieves the state-of-the-art qualitative evaluation and quantitative metrics on some challenging datasets. Experimental results show that our method can be generalized to various models for semantic image synthesis. Our code is available at https://github.com/WangZC525/MUSH.

Keywords: Image synthesis · Semantic information · Multi-scale hierarchical architecture.

1 Introduction

The progress of generative adversarial network (GAN) [7] has promoted the development of image generation technology. However, it is still challenging to generate photorealistic images from input data. In this paper, we focus on semantic image synthesis, which aims to translate semantic label masks to photo-realistic images.

Previous methods [13, 39] directly feed the semantic label masks to an encoder-decoder network. Inspired by AdaIN [10], SPADE [27] uses spatially-adaptive normalization to make semantic information control the generation in normalization layers. This method achieved great success in semantic image synthesis and has been further improved by many recent methods [35, 4, 14, 50, 38, 26, 22].

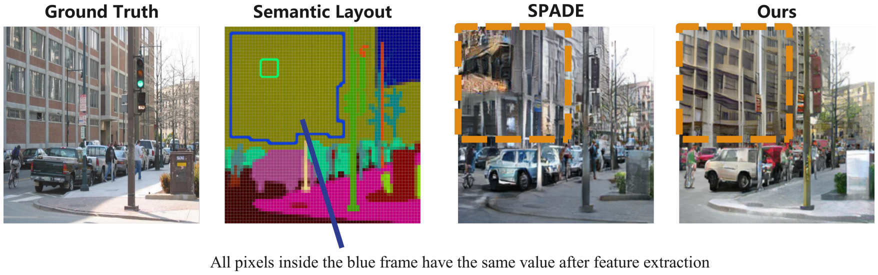

Ground Truth
Semantic Layout
SPADE
Ours
All pixels inside the blue frame have the same value after feature extraction

Fig. 2. An example of consequences of restricted receptive fields in SPADE-based models. We take a semantic label map downsampled to 64x64 in the figure for example. The SPADE-based models have a receptive field of 5x5 (marked with green frame) for feature extraction of semantic layouts. Then in the layout of the building, all pixels inside each blue frame have exactly the same value after convolution, and the model cannot distinguish the relative positions of these pixels in the corresponding objects, leading to bad performance of image synthesis.

However, the receptive fields for semantic feature extraction in these methods are limited, which fails to effectively capture multi-scale information and is unable to distinguish the relative position of each pixel inside its category area, leading to bad performance especially on classes with large areas. CLADE [34, 35] gets similar performance by replacing the semantic feature extraction module of

SPADE by directly mapping each label of the semantic map to the corresponding parameters, which further proves that the semantic feature extraction module in SPADE cannot effectively extract the multi-scale spatial information of the relative positions of pixels inside their categories(see Fig.2).

In addition, the discriminator in previous work [14, 50, 38, 26, 22] takes the concatenation of the image and the semantic label map in channel direction as the input. However, the features of them are added after the first layer of convolution, which can no longer be separated. Moreover, the label map is input only at the beginning of the network in the way of channel concatenation, so that its information can be easily lost in normalization layers, the above problems make it difficult for the network to work according to the label map.

To address above issues, in this paper, we propose a model that utilizes **mu**lti-**s**cale **h**ierarchical architecture to extract semantic features, named MUSH. The architecture is used in both the generator and the discriminator. In the generator, it helps capture specific features of each pixel according to its relative position to improve the performance on classes with large areas. Meanwhile, we merge a method in which semantic features are extracted by using individual parameters for each class with it, so as to achieve good performance on classes with small areas. In the discriminator, we use two different networks to extract semantic mask features and real images respectively. Thus, the discriminator can make better use of the semantic information to distinguish between real and generated images. Because of these strategies, the quality of generated images is significantly improved (see Fig.1).

The contributions of this paper are summarized as follows: (1) We apply a network that can effectively extract the multi-scale features of the semantic map to the generator, so that the generator can recognize the relative positions of all pixels in each object and refine the structure of the generated objects. (2) We propose a novel approach to merge two methods of semantic feature extraction in the generator. (3) We propose a discriminator that extracts the features of the semantic map and the image separately, so as to better discriminate between real and generated images according to the semantic map. (4) With the proposed MUSH, we have achieved better experimental results than the state-of-the-art methods. And we apply our method to GroupDNet and INADE to verify that it can be generalized to various models and improve their performance.

2 Related Work

Generative adversarial networks (GANs) [7] have achieved great success in image generation. It contains two parts: generator and discriminator, which are respectively used to generate images and distinguish between real and fake images. CGAN [23] is proposed based on GAN. It generates images according to restricted input. Our work focuses on CGANs that do semantic image synthesis, where the input is the semantic label map.

Semantic image synthesis takes semantic label maps as inputs, and synthesize photorealistic images according to the maps. Many GAN-based semantic image synthesis models [11, 16, 17, 20, 44, 47, 48] have been proposed recently. Pix2pix [13] and pix2pixHD [39] used an encoder-decoder generator and took semantic maps as inputs directly. CRN [4] refined the output image by a deep cascaded network. SPADE [27] replaced batch normalization [12] in pix2pixHD with spatially adaptive normalization, which has achieved great success in semantic image synthesis. Many approaches have been proposed based on SPADE, such as GroupDNet [50], LGGAN [38], TSIT [14], SEAN [49], INADE [33], OASIS [30], SESAME [26] etc. GroupDNet used group convolution in the generator to implement semantically multi-modal image synthesis. LGGAN proposed a network to generate areas of each category separately. TSIT used a stream to extract features of semantic label maps. SEAN found a better way to insert style information to the network based on SPADE. INADE made the model be adaptive to instances. OASIS trained the network with only adversarial supervision. SESAME helped add, manipulate or erase objects. Different from the above models, our work focuses on the extension of the receptive field for semantic layout feature extraction. Compared with problems that the above models targeted on, the limitation of the receptive field for semantic feature extraction is more serious and easy to be ignored.

Encoder-decoder is a popular structure for deep learning in recent years. It has been applied to many tasks of computer vision. Our semantic map feature extraction network is based on U-Net [29], which is a special encoder-decoder network. Feature maps in its decoder concatenate those the same size as them in encoder by channel, so that the decoder does not forget the encoding information. In addition, due to the basic structure of the encoder-decoder, decoded feature maps contain multi-scale features. The above advantages are what we need for semantic feature extraction. So we propose a semantic feature extraction network based on U-Net.

3 Method

We propose a novel semantic image synthesis model, MUSH. The model is trained adversarially and contains modules with multi-scale hierarchical architectures for semantic feature extraction. In the generator, the module helps image synthesis according to semantic features in various scales. A method using individual parameters for each class is merged to it. In the discriminator, the module helps distinguish between real and generated images according to semantic maps.

3.1 Overall structure of MUSH generator

The generator takes noise as input, and transforms it into an image by convolution, normalization and upsampling. Semantic information are added in the normalization layers. To avoid the limitation of the receptive field for semantic feature extraction and get better feature extraction abilities especially for classes with large areas, we propose a multi-scale hierarchical semantic feature extraction network and multi-scale feature adaptive normalizations (MSFA-Norm). We

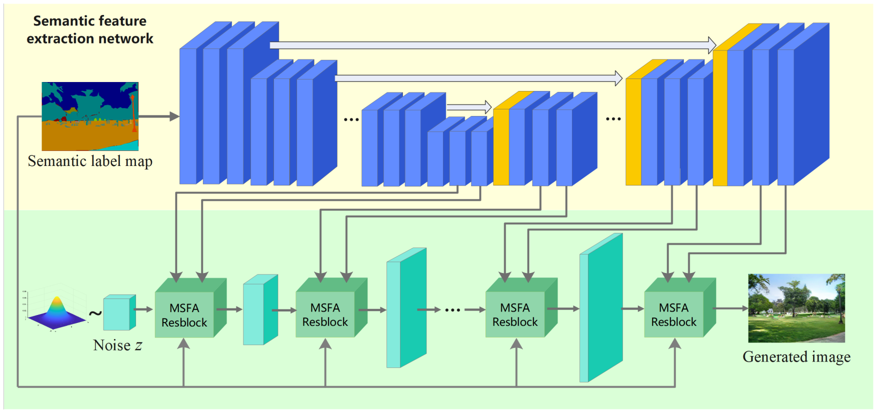


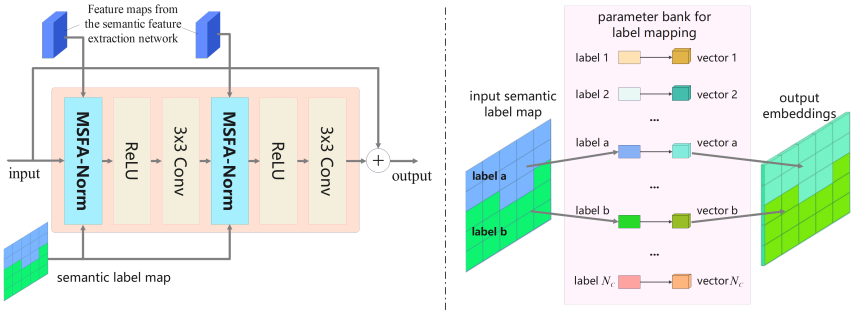
Fig. 3. The overall framework of MUSH generator. The network transforms noise into an image through MSFA-Norm residual blocks. In MSFA residual blocks, semantic information controls the procedure of image generation. The semantic feature extraction network is built based on encoder-decoder structure. Feature maps in the decoder will be input into MSFA residual blocks.

adopt the architecture of resnet blocks [8], pack convolution layers and MSFA-Norm layers into MSFA residual blocks as the main components of the generator (see Fig.3).

The multi-scale hierarchical architecture for semantic feature extraction is based on U-Net, which is an encoder-decoder structure. In encoding stage, the network uses multiple convolutional layers and downsampling layers to get a small-sized encoded feature map, which makes each pixel in the map contain information of a large area. In decoding stage, the network takes the encoded map as input, and outputs multi-scale feature maps from different layers. Therefore the network contains different levels. Each level processes feature maps in a specific size, which builds a multi-scale hierarchical architecture. Additionally, feature maps in the encoder are concatenated to those at the corresponding locations in the decoder with the same channel number to retain the encoding information. Feature maps of each level in the decoder are fed to MSFA-Norm.

3.2 MSFA residual block and MSFA-Norm

The generator uses MSFA residual blocks (see Fig.4 (a)) and upsampling alternatively to transform noise into image. MSFA-Norms in MSFA residual blocks do normalization and control image generation with semantic information, so appropriate form of semantic information should be input here. Since feature maps obtained from the feature extraction network already contains the information of the semantic label map, it seems reasonable to input them into MSFA-Norms without the semantic label map itself. However, although the encoder-decoder extracts features with large receptive fields to achieve good feature extraction results on semantic categories with large areas, it can hardly perform well on



(a) MSFA residual block

(b) Guided Sampling

Fig. 4. The illustration diagrams of the MSFA residual block (a) and the guided sampling operation (b) we use in the block. MSFA residual blocks contains MSFA-Norms, ReLU layers and convolutional layers. In addition to the output of the last layer, the MSFA-Norm also takes the semantic label map and the feature map from the semantic feature extraction network as input. The guided sampling module contains a trainable parameter bank to store a corresponding vector for each label. It replaces each pixel in the semantic label map with the vector corresponding to it during calculation.

semantic categories with small areas. Because of the imbalance of sample quantities for different classes, the network training will be dominated by semantic classes with large areas, leading to classes with small areas being ignored.

Therefore, in the MSFA-Norm (see Fig.5), we add another method to extract semantic features by using individual parameters for each class and combine it to the above mentioned method. Referring to the guided sampling (see Fig.4 (b)) operation in CLADE [34], we maintain a parameter bank which contains a trainable vector for each category. To do the guided sampling operation, we map each category in the semantic label map to the corresponding vector to get the feature map of the input so that the operation is adaptive to different semantic categories. In this approach, we obtain features by individual parameters for each class and avoid training perference for classes with large areas.

However, how to combine these two feature extraction methods? We propose a novel method based on attention values to solve this problem. We obtain a weighted average of the two methods by taking the calculated attention values as the weights of them. For the calculation of the attention values, since performance of these methods on each pixel mainly depends on the category the pixel belongs to, we also use guided sampling to map the semantic layouts to get the attention maps of the two methods. These two maps contains all attention values of the two methods on all pixels. Attention values of different categories vary on these maps.

For the ways in which the two methods influence the input data, we use affine transformation. The multi-scale semantic feature extraction results will be convoluted twice to obtain γ_1 and β_1, which are multiplied and added to the input respectively. For the other method, γ_2 and β_2 are obtained through guided

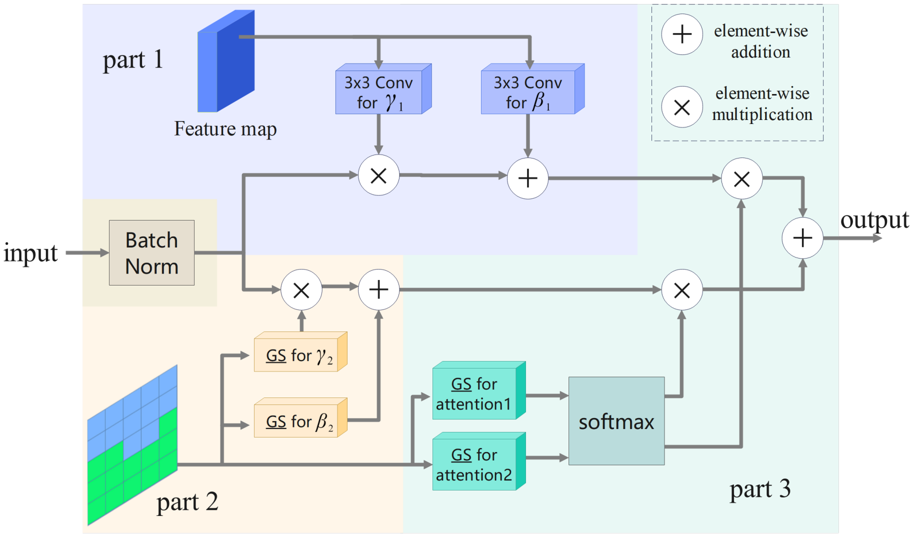


Fig. 5. Structure of MSFA-Norm. GS refers to the guided sampling operation in Fig.4 (b). The feature map in part 1 refers to features obtained from the feature extraction network. Part 1 illustrates how multi-scale features influence the input, and part 2 illustrates how semantic features obtained by guided sampling influence the input. Part 3 shows the process of mergence of results from part 1 and part 2.

sampling, and the affine transformation calculation is the same as above. In addition, before all of these, the input of MSFA-Norms will be batch normalized first.

Let W, H, C and N be the width, height, the number of channels of the feature map to be fed into a MSFA-Norm and the batch size respectively. Value at pixel $p^{n,c,w,h}$ $(n \in N, c \in C, w \in W, h \in H)$ of the feature map will be transformed into $p'^{n,c,w,h}$ after MSFA-Norms, which is expressed as follows:

$$\begin{aligned} p'^{n,c,w,h} = att_1^{n,w,h} \cdot \left(\gamma_1^{n,c,w,h} \cdot \frac{p^{n,c,w,h} - \mu^c}{\sigma^c} + \beta_1^{n,c,w,h}\right) + \\ att_2^{n,w,h} \cdot \left(\gamma_2^{n,c,w,h} \cdot \frac{p^{n,c,w,h} - \mu^c}{\sigma^c} + \beta_2^{n,c,w,h}\right) \end{aligned} \tag{1}$$

Where μ^c and σ^c are the mean and standard deviation of the values in channel c. They are used for batch normalization and expressed as:

$$\mu^c = \frac{1}{NHW} \sum_{nhw} p^{n,c,w,h}, \tag{2}$$

$$\sigma^c = \sqrt{\frac{1}{NHW} \sum_{nhw} \left(p^{n,c,w,h}\right)^2 - \left(\mu^c\right)^2}. \tag{3}$$

att_1 and att_2 refer to attention maps of the multi-scale feature extraction method and the guided sampling method respectively.

Overall, In MSFA-Norm, we use the multi-scale feature extraction results which perform well on semantic classes with large areas. At the same time, the features obtained by guided sampling are added, which has better results on classes with small areas. The two methods are combined through the attention mechanism to achieve good performance on categories in various sizes of areas.

3.3 MUSH discriminator

Previous works [39, 27] use multi-scale PatchGAN discriminator. The image and the semantic layout are concatenated by channel and directly input into the discriminator. But it is hard to distinguish whether information is from images or semantic layouts after the convolutional operation. In addition, the images and the semantic layouts have different scales, which causes calculation deviation after normalization and makes the network lose some semantic information. So in fact, the discriminator ignores much semantic information. To enable the discriminator to separate and extract features of the image and the semantic map while retaining the semantic information in the deep network, we propose a new discriminator.

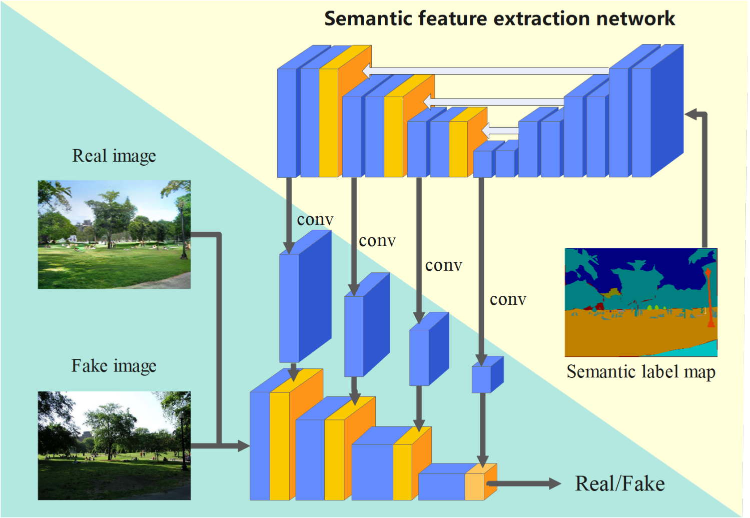


Fig. 6. Structure of MUSH discriminator. The semantic label map and image are input separately into the discriminator. We use a network similar to the semantic feature extraction network in the generator here to produce multi-scale feature maps. The decoded semantic features of each layer are concatenated to the features at all levels of the image, so that input of all convolution layers for images also contains features of the semantic label map.

The discriminator no longer uses the concatenation of image and condition as input like CGAN, but extracts the features of the semantic map separately(see Fig.6). The extracted features at different levels are concatenated to the feature maps at the corresponding levels of image feature extraction, so that each convolutional layer of the discriminator takes both image features and semantic features as input. Therefore the deep network will not forget or discard the semantic information. Finally, the discriminative result will be more related to the semantic information.

The feature extraction network of semantic map in the discriminator is similar to the network described in 3.1. But it uses fewer layers and convolutional channels.

3.4 Training scheme and loss functions

Similar to the original GAN training, we train the discriminator and generator alternately. For the discriminator, we use hinge loss [19, 43] to train referring to previous works [27]. For the generator, most previous works [27, 21, 38, 49, 50] use adversarial loss, GAN feature matching loss and perceptual loss for training. The loss function we use in generator is similar to the above, but the GAN feature matching loss is removed. GAN feature matching loss sets the difference between features of the generated image and the real image as the loss. However, our generator uses multi-scale hierarchical architecture for semantic feature extraction to generate images with finer details, while the detailed representation of each object is diverse and not unique. Training by feature difference may make the model parameters vibrate among different samples, so that the model can be difficult to converge. The training loss functions for the discriminator L_D and for the generator L_G are as follows,

$$L_D = -\mathbb{E}[min(-1 + D(x), 0)] - \mathbb{E}[min(-1 - D(G(z, m)), 0)], \tag{4}$$

$$L_G = -\mathbb{E}[D(G(z, m))] + \lambda_p L_p(G(z, m), x) \tag{5}$$

Where m, x and z refer to the semantic label map, a real image and the input noise of generator respectively. $G(z, m)$ denotes the image synthesized by the generator with noise z and the semantic map as input. $L_p(G(z, m), x)$ denotes the perceptual loss [15], which is used to minimize the VGG19 [32] feature difference between the generated image and the real image. λ_p refers to the weight of perceptual loss.

4 Experiments

4.1 Implementation details

We apply the spectral norm [24] to layers in both generator and discriminator. The learning rates of generator and discriminator are 0.0001 and 0.0004 respectively. We adopt Adam optimizer [18] with $\beta 1 = 0$ and $\beta 2 = 0.999$. We train our model on V100 GPUs. The weight of perceptual loss λ_p in the loss function is 10.

4.2 Datasets and metrics

We conduct experiements on COCO-Stuff [2], ADE20K [45], ADE20K-outdoor and Cityscapes [5]. Many previous works [36, 4, 47, 1, 37, 28, 26, 6, 22] have experiemented on these datasets. COCO-Stuff has 182 semantic categories, containing 118,000 training images and 5,000 validation images. Its huge and diverse content makes it very challenging. ADE20K contains 20,210 training images and 2,000 validation images, 150 categories in total. It is a challenging dataset, too. ADE20K-outdoor is a subset of ADE20K. Cityscapes is a dataset of street scene images. Its training set and validation set contain 3,000 and 500 images respectively. All images are in high resolution, which makes it suitable for testing high resolution image synthesis of models. We use the Fréchet Inception Distance (FID) [9, 31] to measure the distribution distance between the model generated images and the real images. Additionally, we use mean Intersection-over-Union (mIOU) and pixel accuracy (accu) measured by the state-of-the-art image segmentation networks for each dataset: DeepLabV2 [3, 25] for COCO-Stuff, UperNet101 [46, 40] for ADE20K, and DRN-D-105 [42, 41] for Cityscapes.

4.3 Qualitative results

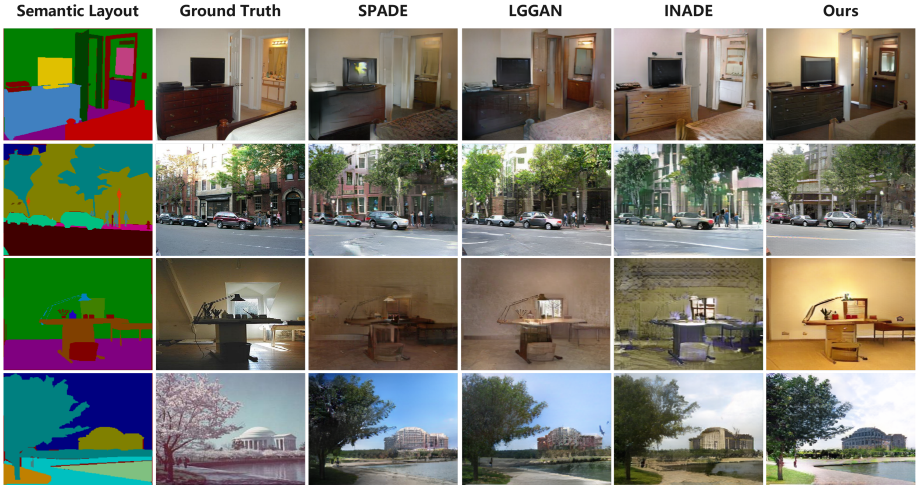


Fig. 7. Comparison of MUSH with other methods on ADE20K

As shown in Fig.7, we compare our results with the state-of-the-art approach SPADE and two popular SPADE-based approaches LGGAN and INADE. Ours have finer details, clearer eages and less generation failure (For example, the furniture in the first row has straight edges. The building in the second and the fourth row are less blurred. The wall in the third row is more clear). In addition, the model is able to consider the internal representation differentiation of different locations of complex objects, which makes the objects better generated in all parts.

4.4 Quantitative results

We use popular semantic image synthesis models in recent years as baselines, including CRN, pix2pixHD, SPADE, GroupDNet, LGGAN, TSIT and INADE. SPADE is the state-of-the-art approach.

Table 1. Quantitative comparison of MUSH with other methods. Bold denotes the best performance. For the mIoU and accu, higher is better. For the FID, lower is better. Results of GroupDNet, LGGAN, TSIT and INADE are collected by running the evaluation on our machine.

	ADE20K			ADE20K-outdoor			Cityscapes			COCO-Stuff		
	mIoU	Accu	FID	mIoU	Accu	FID	mIoU	Accu	FID	mIoU	Accu	FID
CRN	22.4	68.8	73.3	16.5	68.6	99.0	52.4	77.1	104.7	23.7	40.4	70.4
pix2pixHD	20.3	69.2	81.8	17.4	71.6	97.8	58.3	81.4	95.0	14.6	45.8	111.5
SPADE	38.5	79.9	33.9	30.8	82.9	63.3	62.3	81.9	71.8	37.4	67.9	22.6
GroupDNet	28.3	74.7	42.0	n/a	n/a	n/a	62.5	82.2	50.2	n/a	n/a	n/a
LGGAN	38.8	80.6	32.2	n/a	n/a	n/a	64.4	82.9	55.8	n/a	n/a	n/a
TSIT	37.2	80.3	33.4	n/a	n/a	n/a	64.0	83.7	55.5	n/a	n/a	n/a
INADE	37.7	79.9	33.5	n/a	n/a	n/a	61.2	81.9	49.9	n/a	n/a	n/a
OASIS	45.0	83.6	30.7	**36.2**	85.8	55.5	**67.7**	86.3	49.3	**44.5**	**71.4**	**16.8**
SESAME	**45.7**	**84.9**	31.4	n/a	n/a	n/a	65.8	84.2	51.6	n/a	n/a	n/a
Ours	39.0	82.5	**30.3**	33.7	**86.7**	**55.2**	66.3	**87.2**	**48.0**	36.7	69.3	21.6

As shown in Table 1, our method outperforms most baselines expect OASIS and SESAME in almost all metrics on each dataset. It achieves great results especially on small datasets such as Cityscapes and ADE20K-outdoor. However, it does not get better results than OASIS and SESAME on COCO-Stuff or ADE20K. Because we used a relatively lightweight semantic feature extraction network in the experiment so that for big datasets, it does not have sufficient parameters to extract all global features of semantic label maps with so many categories. An increase in its scale can help achieve better results. In spite of these, our model in this scale gets best FID on most datasets except COCO-Stuff and is still competitive with these models. In addition, we use SPADE as the backbone of our method in the experiment. A better backbone will help improve the performance.

4.5 Generalization ability to SPADE-Based methods

We also carried out the improvement experiments of our methods on recent semantic image synthesis models with other functions. We choose two representative method: GroupDNet [50] and INADE [33]. GroupDNet replaces convolution

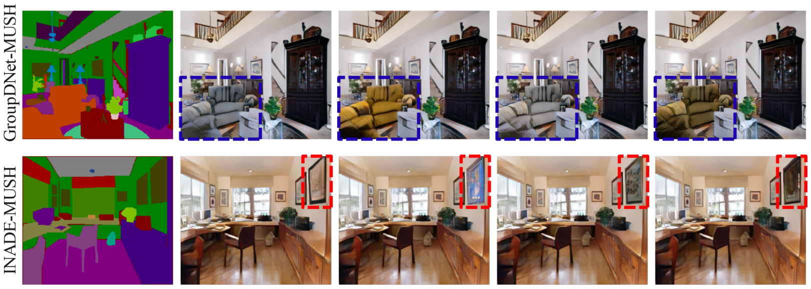


Fig. 8. After being applied to other models, our method can inherit their functions. MUSH has great performance on multi-modal image synthesis (first row, various sofas can be synthesized) by being applied to GroupDNet, and can implement instance-level multi-model image synthesis (second row) by being applied to INADE.

layers in SPADE encoder and generator with group convolution layers, so as to implement semantically multi-modal image synthesis. INADE uses random values from parametric probability distributions in denormalization to implement instance-level multi-model image synthesis. Both of them achieve good performance and implement new functions.

Table 2. Generalization ability test of MUSH on GroupDNet and INADE. Bold denotes the best performance. All results are collected by running the evaluation on our machine.

	ADE20K			Cityscapes		
	mIoU↑	Accu↑	FID↓	mIoU↑	Accu↑	FID↓
GroupDNet	28.3	74.7	42.0	62.5	82.2	50.2
GroupDNet-MUSH	**36.7**	**78.8**	**34.8**	**63.0**	**82.7**	**49.0**
INADE	37.7	79.9	33.5	61.2	81.9	49.9
INADE-MUSH	**38.6**	**81.5**	**30.9**	**64.5**	**85.3**	**49.5**

We apply MUSH's multi-scale hierarchical semantic feature extraction modules to GroupDNet and INADE. Similar to what is described in section 3.2, we use guided sampling to calculate attention values of multi-scale feature extraction methods and their original denormalization methods to obtain the output of normalization layers. The new models not only inherit their functions (see Fig.8), but also achieve better performance (see Table 2), especially for GroupDNet on ADE20K. The results show that our method has great generalization ability.

4.6 Ablations

We conduct the ablation experiments on ADE20K and Cityscapes. These are ablation configurations on generator architecture and discriminator architecture. The experimental results and our analysis are as follows.

Table 3. Ablation on generator architecture. Bold denotes the best performance.

	ADE20K		Cityscapes	
Generator architecture	mIoU↑	FID↓	mIoU↑	FID↓
MUSH	**39.0**	**30.3**	**66.3**	**48.0**
MUSH w/o guided sampling	37.0	30.9	63.4	49.5
SPADE	37.2	32.5	62.8	58.6

Ablation on the generator architecture. We train some alternative generators. The results are shown in Table 3. Compared to SPADE generator, MUSH generator performs better on both mIoU and FID. We also train a generator without guided sampling, which means that the guided sampling part for semantic feature extraction and the attention calculation part are eliminated, so that the MSFA controls image generation only by multi-scale semantic features. We find that the network performs worse, especially on mIoU. This shows that the multi-scale hierarchical architecture fails to extract features of classes with small areas because mIoU calculates the average results of all classes, while most classes have small areas in images of both two datasets, leading to low mIoU of its results on them. This also verifies that the method of guided sampling and the approach we add it here improve the performance.

Table 4. Ablation on discriminator architecture. Bold denotes the best performance.

	ADE20K		Cityscapes	
discriminator architecture	mIoU↑	FID↓	mIoU↑	FID↓
MUSH	**39.0**	**30.3**	**66.3**	**48.0**
MUSH w/ GAN feature match loss	38.0	31.4	63.2	52.7
MUSH w/o VGG loss	33.2	40.5	59.3	63.4
SPADE	38.5	31.2	64.7	52.1

Ablation on the discriminator architecture. As shown in Table 4, compared to the MUSH generator, MUSH discriminator improves less performance but still performs better than SPADE discriminator. The experimental results also show that the GAN feature matching loss in MUSH will degrade the performance. When GAN feature matching loss is used, different samples of images with diversity will make the generator network converge in different directions in training, resulting in difficulty in model convergence and blurred areas in generated images. We also get the results that VGG loss is essential in our network. It is difficult to generate so complex images with adversarial training only, so it is reasonable that we use features from a pretrained model to guide learning.

5 Conclusions

In this paper, we propose MUSH, a semantic image synthesis method that extracts semantic features with a multi-scale hierarchical architecture. The feature extraction network for semantic label maps can calculate an unique value for each pixel, which benefits generation of classes with large areas. We also merge a semantic feature extraction method in which individual parameters are used for each class with it in order to get better results on classes with small areas. Because of these, MUSH generator performs well on various classes. The MUSH discriminator extracts features of the semantic label map and the image separately, which makes it better discriminate between real and fake images according to the semantic label map. MUSH achieves better results than the state-of-the-art approaches and can be generalized to various models to improve their performance. However, for large datasets like COCO-Stuff, it is hard for the proposed network to extract all features of semantic maps. We believe this is a promising research area.

Acknowledgements This work was supported by the National Key Research and Development Program of China under Grant 2018YFB2100801.

References

1. Almahairi, A., Rajeshwar, S., Sordoni, A., Bachman, P., Courville, A.: Augmented cyclegan: Learning many-to-many mappings from unpaired data. In: International Conference on Machine Learning. pp. 195–204. PMLR (2018)
2. Caesar, H., Uijlings, J., Ferrari, V.: Coco-stuff: Thing and stuff classes in context. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1209–1218 (2018)
3. Chen, L.C., 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 and machine intelligence **40**(4), 834–848 (2017)
4. Chen, Q., Koltun, V.: Photographic image synthesis with cascaded refinement networks. In: Proceedings of the IEEE international conference on computer vision. pp. 1511–1520 (2017)
5. 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: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3213–3223 (2016)
6. Dundar, A., Sapra, K., Liu, G., Tao, A., Catanzaro, B.: Panoptic-based image synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8070–8079 (2020)

7. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in neural information processing systems **27** (2014)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
9. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017)
10. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: Proceedings of the IEEE international conference on computer vision. pp. 1501–1510 (2017)
11. Huang, X., Liu, M.Y., Belongie, S., Kautz, J.: Multimodal unsupervised image-to-image translation. In: Proceedings of the European conference on computer vision (ECCV). pp. 172–189 (2018)
12. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. pp. 448–456. PMLR (2015)
13. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1125–1134 (2017)
14. Jiang, L., Zhang, C., Huang, M., Liu, C., Shi, J., Loy, C.C.: Tsit: A simple and versatile framework for image-to-image translation. In: European Conference on Computer Vision. pp. 206–222. Springer (2020)
15. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European conference on computer vision. pp. 694–711. Springer (2016)
16. Karacan, L., Akata, Z., Erdem, A., Erdem, E.: Learning to generate images of outdoor scenes from attributes and semantic layouts. arXiv preprint arXiv:1612.00215 (2016)
17. Karacan, L., Akata, Z., Erdem, A., Erdem, E.: Manipulating attributes of natural scenes via hallucination. ACM Transactions on Graphics (TOG) **39**(1), 1–17 (2019)
18. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (Poster) (2015)
19. Lim, J.H., Ye, J.C.: Geometric gan. arXiv preprint arXiv:1705.02894 (2017)
20. Liu, M.Y., Breuel, T., Kautz, J.: Unsupervised image-to-image translation networks. Advances in neural information processing systems **30** (2017)
21. Liu, X., Yin, G., Shao, J., Wang, X., et al.: Learning to predict layout-to-image conditional convolutions for semantic image synthesis. Advances in Neural Information Processing Systems **32** (2019)
22. Long, J., Lu, H.: Generative adversarial networks with bi-directional normalization for semantic image synthesis. In: Proceedings of the 2021 International Conference on Multimedia Retrieval. pp. 219–226 (2021)
23. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)
24. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. In: International Conference on Learning Representations (2018)
25. Nakashima, K.: Deeplab-pytorch. https://github.com/kazuto1011/deeplab-pytorch (2018)

26. Ntavelis, E., Romero, A., Kastanis, I., Gool, L.V., Timofte, R.: Sesame: Semantic editing of scenes by adding, manipulating or erasing objects. In: European Conference on Computer Vision. pp. 394–411. Springer (2020)
27. Park, T., Liu, M.Y., Wang, T.C., Zhu, J.Y.: Semantic image synthesis with spatially-adaptive normalization. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2337–2346 (2019)
28. Qi, X., Chen, Q., Jia, J., Koltun, V.: Semi-parametric image synthesis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8808–8816 (2018)
29. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
30. Schonfeld, E., Sushko, V., Zhang, D., Gall, J., Schiele, B., Khoreva, A.: You only need adversarial supervision for semantic image synthesis. In: International Conference on Learning Representations (2020)
31. Seitzer, M.: pytorch-fid: FID Score for PyTorch. https://github.com/mseitzer/pytorch-fid (August 2020), version 0.2.1
32. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
33. Tan, Z., Chai, M., Chen, D., Liao, J., Chu, Q., Liu, B., Hua, G., Yu, N.: Diverse semantic image synthesis via probability distribution modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7962–7971 (2021)
34. Tan, Z., Chen, D., Chu, Q., Chai, M., Liao, J., He, M., Yuan, L., Hua, G., Yu, N.: Efficient semantic image synthesis via class-adaptive normalization. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021)
35. Tan, Z., Chen, D., Chu, Q., Chai, M., Liao, J., He, M., Yuan, L., Yu, N.: Rethinking spatially-adaptive normalization. arXiv preprint arXiv:2004.02867 (2020)
36. Tang, H., Bai, S., Sebe, N.: Dual attention gans for semantic image synthesis. In: Proceedings of the 28th ACM International Conference on Multimedia. pp. 1994–2002 (2020)
37. Tang, H., Qi, X., Xu, D., Torr, P.H., Sebe, N.: Edge guided gans with semantic preserving for semantic image synthesis. arXiv preprint arXiv:2003.13898 (2020)
38. Tang, H., Xu, D., Yan, Y., Torr, P.H., Sebe, N.: Local class-specific and global image-level generative adversarial networks for semantic-guided scene generation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7870–7879 (2020)
39. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional gans. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8798–8807 (2018)
40. Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J.: Unified perceptual parsing for scene understanding. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 418–434 (2018)
41. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. In: International Conference on Learning Representations (ICLR) (2016)
42. Yu, F., Koltun, V., Funkhouser, T.: Dilated residual networks. In: Computer Vision and Pattern Recognition (CVPR) (2017)
43. Zhang, H., Goodfellow, I., Metaxas, D., Odena, A.: Self-attention generative adversarial networks. In: International conference on machine learning. pp. 7354–7363. PMLR (2019)

44. Zhao, B., Meng, L., Yin, W., Sigal, L.: Image generation from layout. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8584–8593 (2019)
45. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 633–641 (2017)
46. Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., Torralba, A.: Semantic understanding of scenes through the ade20k dataset. International Journal on Computer Vision (2018)
47. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2223–2232 (2017)
48. Zhu, J.Y., Zhang, R., Pathak, D., Darrell, T., Efros, A.A., Wang, O., Shechtman, E.: Toward multimodal image-to-image translation. Advances in neural information processing systems **30** (2017)
49. Zhu, P., Abdal, R., Qin, Y., Wonka, P.: Sean: Image synthesis with semantic region-adaptive normalization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5104–5113 (2020)
50. Zhu, Z., Xu, Z., You, A., Bai, X.: Semantically multi-modal image synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5467–5476 (2020)