Mask-based Style-Controlled Image Synthesis Using a Mask Style Encoder

Jaehyeong Cho, Wataru Shimoda and Keiji Yanai
Department of Informatics, The University of Electro-Communications, Tokyo, Japan
Email: {cho,shimoda-k,yanai}@mm.inf.uec.ac.jp

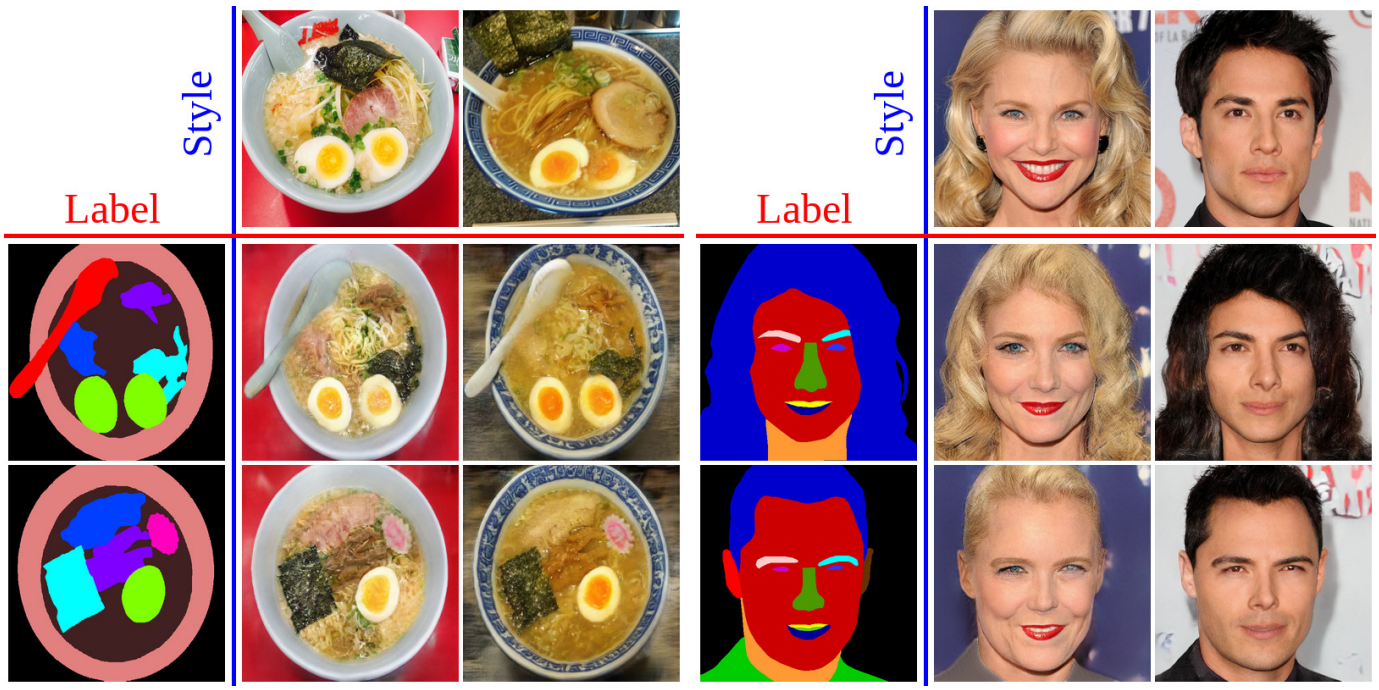


Fig. 1. The proposed method on mask-based style-controlled image synthesis can generate realistic images where the styles of each mask element are transferred to the corresponding region mask elements. In the left figure, the images of Ramen noodles are generated based on segmentation masks (in the leftmost column) and style images (in the top row). The color of tables and soups, and the texture of bowls are successfully transferred. In the right figure, the facial images are generated from the facial region masks with the mask style of each of the facial parts.

***Abstract*—In recent years, the advances in Generative Adversarial Networks (GANs) have shown impressive results for image generation and translation tasks. In particular, the image-to-image translation is a method of learning mapping from a source domain to a target domain and synthesizing an image. Image-to-image translation can be applied to a variety of tasks, making it possible to quickly and easily synthesize realistic images from semantic segmentation masks. However, in the existing image-to-image translation method, there is a limitation on controlling the style of the translated image, and it is not easy to synthesize an image by controlling the style of each mask element in detail. Therefore, we propose an image synthesis method that controls the style of each element by improving the existing image-to-image translation method. In the proposed method, we implement a mask style encoder that extracts style features for each mask element. The extracted style features are concatenated to the semantic mask in the normalization layer, and used the style-controlled image synthesis of each mask element. In the experiments, we performed style-controlled images synthesis using the datasets consisting of semantic segmentation masks and real images. The results show that the proposed method has excellent performance for style-controlled images synthesis for each element.**

I. Introduction

There are various reasons for synthesizing and editing images, and synthesizing and editing images is intended to make images more attractive. For example, in recent years, a large number of images have been uploaded to social network services (SNS) on the Web, such as Twitter and Instagram, and the amount is increasing everyday. When uploading images on the Web, some users edit images to make them more attractive images using some photo editing tools. However, editing images is not an easy task which requires special skills and a lot of time.

On the other hand, the accuracy and quality of various researches and tasks using images have been greatly improved by the development of deep learning. In particular, Generative Adversarial Networks (GANs) [1] using deep learning are used as a powerful framework for generating and translating high-quality images in various researches. Image-to-image transformation is a method of conditional image synthesis that learns the mapping from a source domain to a target domain. This method can be applied to various tasks including image synthesis from semantic segmentation masks. Segmentation-

mask-based image-to-image translation [2]–[5] can be applied to generation and editing of new contents. The advantage of this method is that real image synthesis can be performed quickly and easily while controlling the shape of the synthesized image. However, synthesizing images while controlling the style of each element remains a difficult task.

In this paper, we propose a style-controlled image synthesis method for each make element using a mask style encoder. We adopt a SPADE-based architecture [5] as a base architecture on image translation. The SPADE-based method can synthesize a realistic image from a semantic segmentation mask image. However, it cannot specify the style of each of the mask regions. For example, it cannot change only hair color with keeping the style of other parts unchanged when synthesizing a facial image. To solve this problem, we propose a mask style encoder which extracts a style feature of each mask element from a given style image. To make image synthesis take into account mask style features, we modify a SPADE-based image generator by providing mask style features as well as region mask information by concatenating them. We train the SPADE-based generator with the mask style encoder, and perform style-controlled image synthesis of mask elements.

In the experiment, we train the generator using the images and semantic labels included in the food, face and others image datasets, and perform style-controlled image synthesis of each element as shown in Figure 1. In addition, we compare the results of the existing method and style-controlled image synthesis, and show that the proposed method has better performance in style-controlled image synthesis for each element than the existing method.

Our main contributions are summarized as follows:

- We propose a mask style encoder which extracts mask style features for each element, and the modified SPADE-based generator which can synthesize style-controlled images with the extracted mask style features.
- We demonstrate the performance of the proposed method by comparing the results of style-controlled image synthesis for each element of the proposed method with the existing method.

II. Related works

Generative adversarial networks Generative Adversarial Network (GAN) [1] is an image synthesis method that consists of a generator and a discriminator. A discriminator distinguishes between real and synthesized images, while a generator produces more realistic images that the discriminator cannot distinguish between real and synthesized images. GAN enables a wide variety of applications such as image generation [6]–[10], image inpainting [11]–[14], image manipulation [15]–[17] and super resolution [18]. Conditional GAN (cGAN) [19] enables image synthesize to control attributes by adding condition information to the GAN architecture. For example, it has become possible to perform image synthesize based on category labels [20], image synthesize based on text [21], [22], image synthesize between different domain images [2], [23].

Image-to-image translation Image-to-image translation aims to learn the mapping from a source domain to a target domain. Recently, GAN-based image-to-image translation methods are being studied very actively. Image-to-image translation can divided into two types, paired or supervised image-to-image translation [2]–[5], [24], [25] in which the training samples in both domains have correspondence to each other, and unpaired or unsupervised image-to-image translation [23], [26]–[30] in which the learning samples in both domains do not correspond to each other. For paired image-to-image translation, Isola et al. proposed Pix2pix [2], which performed image-to-image translation using cGAN [19] and U-Net [31] networks. However, Pix2pix could only synthesize images at a low resolution of up to 256×256 sizes. Wang et al. proposed Pix2pixHD [4] that improved on Pix2pix to produce high-resolution images. Park et al. proposed a spatially-adaptive normalization (SPADE) [5] architecture which enables us to synthesize realistic images from segmentation mask images. We explain the detail of SPADE in the next section, since we use the extended version of it as an image generator.

In our work, we focus not only on the translation of segmentation masks into photo-realistic images, but also on the translate of styles of each mask element such as hair, mouth and face in case of human faces. In our proposed method, the generator part is based on the architecture of SPADE, and we introduce a method of image synthesis considering styles of each mask element. We use the datasets that contain pairs of segmentation masks and real images for training.

Note that SEAN [32] is the concurrent work to ours, which proposed region-wise average pooling to inject region-wise style features into SPADE. They proposed SEAN normalization which is a modification of SPADE normalization. Compared to that, our architecture is much simpler since we use SPADE normalization as it is by concatenating a segmentation mask and region-wise style vectors. However, the quality of synthesized images are comparable.

III. SPADE

In this section, we explain the SPADE-based architecture [5] which is the state-of-the-art image translation method from a segmentation mask image to a realistic image. SPADE is the abbreviation of "SPatially-ADaptivE normalization", which is a kind of a conditional normalization. SPADE is similar to batch normalization, in which activations are normalized in the channel-wise manner and then modulated with learned scale and bias parameters. Figure 2 shows the SPADE normalization layers.

Let m be a semantic segmentation mask which is represented as a one-hot label map in the SPADE layers. Let a^i denote the activations of the i-th layer of a deep convolutional network for a batch of N samples. Let C^i be the number of channels in the layer. Let H^i, W^i and C^i be the height and width of the activation map in the layer and the number of channels in the layer. The activation value at site $(n \in N, c \in C^i, h \in H^i, w \in W^i)$ is

$$\gamma^i_{c,h,w}(m)\frac{a^i_{n,c,h,w} - \mu^i_c}{\sigma^i_c} + \beta^i_{c,h,w}(m) \tag{1}$$

where $a^i_{n,c,h,w}$ is the activation at the site before normalization. μ^i_c and σ^i_c are the mean and standard deviation of the

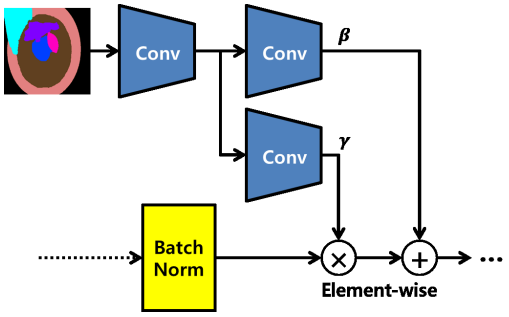


Fig. 2. The structure of SPatially-ADaptivE normalization (SPADE) [5], which generates modulation parameters, γ and β, from a given semantic mask. The generated parameters, γ and β, are multiplied and added to the normalized activations in the element-wise manner.

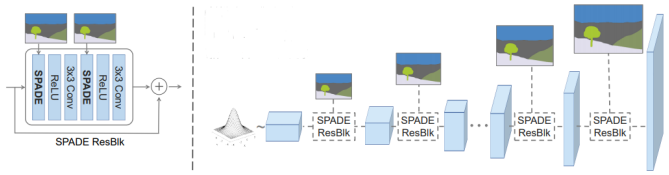


Fig. 3. The architecture of the SPADE-based network. Two SPADE layers are used in each SPADE residual block (SPADE ResBlock). The network consists of the SPADE ResBlocks and the upsampling layers alternately. (The figure is cited from [5].)

activations in channel c:

$$\mu_c^i = \frac{1}{NH^iW^i} \sum_{n,h,w} a^i_{n,c,h,w} \tag{2}$$

$$\mu_c^i = \sqrt{\frac{1}{NH^iW^i} \sum_{n,h,w} \left((a^i_{n,c,h,w})^2 - (\mu_c^i)^2 \right)}. \tag{3}$$

The variables $\gamma^i_{c,h,w}(m)$ and $\beta^i_{c,h,w}(m)$ are the learned modulation parameters of the normalization layer.

The SPADE generator architecture removed the encoder part of the image-to-image translation network used in the existing method [2], [4], and adopted SPADE normalization layers for all the normalization layers of the generator as shown in Figure 3. The modulation parameters of all the normalization layers are learned using the SPADE. The generator is trained using the same multi-scale discriminator and loss function as Pix2pixHD [4], except replacing the least squared loss term [7] with the hinge loss term [33], [34]. The SPADE method enabled us to synthesize high-quality, photo-realistic images from semantic masks. However, there is still a problem in synthesizing images by controlling the style of each mask element. Therefore, we propose a style controlled image synthesis method for each mask element based on the SPADE method in the next section.

IV. Proposed Method

In this section, we explain a mask style encoder which extracts style features from each of the mask elements, and a mask-style-based generator which synthesizes realistic images from segmentation masks with mask style features.

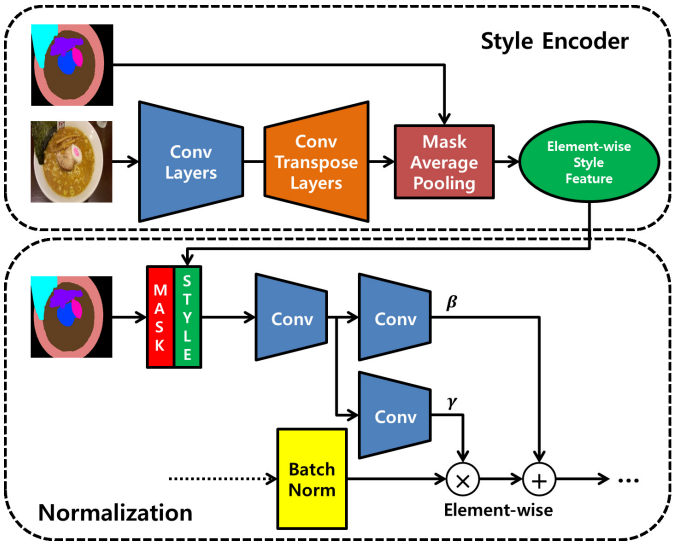


Fig. 4. In the style encoder, a style feature for each element is extracted using a style image and a semantic mask corresponding to the style image. In normalization, the extracted style features are concatenated with the semantic mask and used to generate the modulation parameters γ and β.

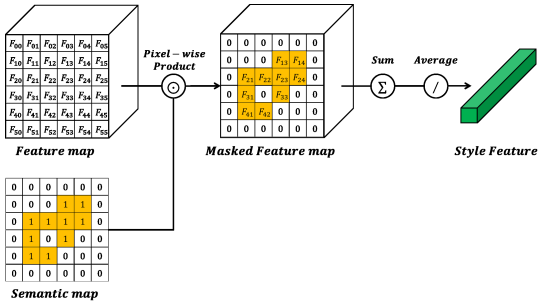


Fig. 5. In the mask averaging pooling, the extracted feature map is multiplied by a semantic mask to obtain a masked feature map. Then, the sum of the masked features is obtained, and the average of the features is calculated using the number of masked pixels to extract style features.

A. Mask Style Encoder

We propose a style encoder which extracts mask style features of region mask elements from given style images for the purpose of synthesizing mask-style-controlled images. The style encoder takes a semantic segmentation mask and a style image, and extracts style features from each of the mask elements of the style image such hair, skin and mouth regions in case of human face.

The upper part of Figure 4 shows the basic architecture of the mask style encoder which consists of an encoder-decoder-style feature extractor and a mask pooling layer. First, a style image is provided to an encoder-decoder network which is composed of convolution layers and transposed convolution layers, and then a style feature map is extracted. Second, the mask average pooling which takes a semantic segmentation mask corresponding to the style image as an auxiliary input is applied to the feature map to extract the style features of each mask element.

We propose a mask average pooling layer which extracts averaged feature vectors of the input feature map over the

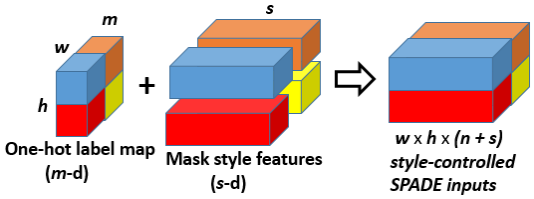


Fig. 6. The input of the style-controlled SPADE which is a $h \times w \times (m+s)$ tensor. It is a concatenation of a one-hot label map of a given segmentation mask and a set of s-d style feature vectors which are broadcasted to the corresponding label locations. m and s represents the number of mask labels and the dimension of mask style vectors, respectively. Note that $h \times w$ represents the size of the feature map where the SPADE modulation is injected.

specific region label as shown in Figure 5. We regard the feature map as an aggregation of feature vectors aligned in the channel direction. The figure shows the mask pooling layer averages feature vectors in the corresponding spatial locations (highlighted in yellow) to the specific region label. By repeating this computation for all the mask labels, we can extract averaged style features of the segmentation mask region elements such as bowl, background and soup regions in case of Ramen noodle images shown in the figure. We use these mask style vectors corresponding to the mask elements for mask-style-based image synthesis. Note that the semantic mask is resized to the same size as the input feature map in the mask pooling layer.

B. Element-Wise Style Controlled Image Synthesis

The extracted mask style features are concatenated with the semantic mask label map, and they are provided to SPADE layers [5] to perform mask-element-wise style-controlled image synthesis as shown in the lower part of Figure 4. Figure 6 shows the way to concatenate the label map represented by a set of one-hot vectors and a set of style feature vectors which are broadcasted to the corresponding label locations. The equation in the style-controlled SPADE layer is as follows:

$$\gamma^i_{c,h,w}(s,m)\frac{a^i_{n,c,h,w}-\mu^i_c}{\sigma^i_c}+\beta^i_{c,h,w}(s,m). \tag{4}$$

where s represents a map of broadcasted mask style features. The architecture of the generator network is the same as the generator of [5] as shown in Figure 3. As a discriminator, we use the multi-scale discriminator used in Pix2pixHD [4].

C. Loss Functions

We use a modified loss function to control the styles extracted from the style encoder for the three loss functions used in Pix2pixHD [4] and SPADE [5] for training the generator.

1) Adversarial loss: Conditional adversarial loss models the conditional distribution of real images through the minimax game. Let G be the generator, D_1, D_2 be the two discriminators, E be a mask style encoder. The Adversarial loss is then calculated as:

$$\min_{G,E}\max_{D_1,D_2}\sum_{k=1,2}\mathcal{L}_{GAN}(G,D_k,E). \tag{5}$$

where the objective function $\mathcal{L}_{GAN}$ uses hinge loss [33], [34] and is given by

$$\begin{aligned}\mathcal{L}_{GAN}(G,D,E) = \\ -\mathbb{E}[min(0,-1+D_k(s,x))] \\ -\mathbb{E}[min(0,-1-D_k(s,G(s,E(s,x))))],\end{aligned} \tag{6}$$

where x is the real image, s is the semantic mask corresponding to x.

2) Feature matching loss: Feature matching loss stabilizes the training as the generator has to produce natural statistics at multiple scales. Let $D^{(i)}_k$ be the i-th layer feature extractor of discriminator D_k. The feature matching loss $\mathcal{L}_{FM}$ is then calculated as:

$$\begin{aligned}\mathcal{L}_{FM}(G,D_k,E) = \\ \mathbb{E}\sum_{i=1}^{T}\frac{1}{N_i}[\|D^{(i)}_k(s,x)-D^{(i)}_k(s,G(s,E(s,x)))\|_1],\end{aligned} \tag{7}$$

where T is the total number of layers, N_i is the number of elements in each layer.

3) Perceptual loss: Perceptual loss helps to synthesize images in more detail using a pretrained VGG network [35]. Let $F^{(i)}$ be the i-th layer of the VGG network, M_i be the number of elements in the VGG network. The perceptual loss $\mathcal{L}_{percept}$ is then calculated as:

$$\begin{aligned}\mathcal{L}_{percept}(G,F,E) = \\ \mathbb{E}\sum_{i=1}^{N}\frac{1}{M_i}[\|F^{(i)}(x)-F^{(i)}(G(s,E(s,x)))\|_1].\end{aligned} \tag{8}$$

4) Total loss: Finally, we train a generator that combines the three loss functions and synthesizes a style-controlled image for each element. The total loss $\mathcal{L}_{total}$ combining the three loss functions is calculated as follows:

$$\begin{aligned}\mathcal{L}_{total}(G,D_k,F,E) = \\ \min_{G,E}\Bigg(\Big(\max_{D_1,D_2}\sum_{k=1,2}\mathcal{L}_{GAN}(G,D_k,E)\Big) \\ +\lambda_1\sum_{k=1,2}\mathcal{L}_{FM}(G,D_k,E)+\lambda_2\mathcal{L}_{percept}(G,F,E)\Bigg).\end{aligned} \tag{9}$$

V. Experiments

We first perform image synthesis using a single style image using the proposed method, and compare the results with the existing baseline method. Then, we extract the style of each mask element from multiple style images, and perform style-controlled image synthesis.

A. Implementation details

Following the work of SPADE [5], we apply the Spectral Norm [33] in both a generator and a discriminator. The learning rates for the generator and discriminator are 0.0001 and 0.0004, respectively [36]. We use the ADAM [37] with $\beta_1 = 0$ and $\beta_2 = 0.9$. We set the weight $\lambda_1 = \lambda_2 = 10$ in Eq.(9).

B. Datasets

We conduct experiments on four kinds of the public datasets, including real images and semantic masks.

- UEC-Ramen555 dataset [38] consists of 555 real ramen images and the corresponding semantic segmentation masks. It has 11 categories of pixel-wise labels and 5 soup sub-category labels.
- CelebAMask-HQ dataset [39] is a dataset derived from CelebA [9], [40], which contains 30,000 high-resolution face images and semantic masks. It has 19 kinds of semantic labels.
- ADE20K dataset [41] consists of 20,210 training images and 2,000 validation images. It has 150 classes of semantic labels.
- Cityscapes dataset [42] consists of 2,975 training images and 500 validation images. It has 35 kinds of semantic classes.

C. Quantitative results

We compare our method with the SPADE model [5] quantitatively, which is currently the state-of-the-art mask-to-image translation method. In order to compare with the baseline method, we perform image synthesis using only a single style image, and report the performance of image synthesis. We use the Fréchet Inception Distance (FID) [36] as a metric to evaluate the results of image synthesis for each method. The FID score uses the Inception-v3 [43] model to calculate the distance between the distribution of the features in real images and synthesis images, and measures the similarity between the distribution of real images and synthesis images. Therefore, a lower FID score indicates better quality.

Table I shows the FID score for the five dataset. For UEC-Ramen555 and Cityscapes, SPADE has a better FID score. However, the results of CelebAMask-HQ and ADE20K show that our method achieved better FID scores than the existing method. Especially in CelebA, we achieved much better FID score. This shows that the the quality of image synthesis is almost equivalent to the original method even when performing style-controlled image synthesis.

D. Qualitative results

In Figure 7, we provide a qualitative comparison with the original SPADE [5]. We see that our method produced better style-controlled image synthesis results than the existing method. For example, the color style of the bowls from the top to the third row, the controlled style of hair and beard from the forth to the sixth row, and the scene styles below the seventh rows look better than the results by the SPADE.

In Figure 8, we show the matrix of the results between five style images and four mask images with two kinds of the dataset, UEC-Ramen555 and CelebAMask-HQ, which indicated the proposed style-controlled image synthesis works very successfully.

In Figure 9, we extracted mask style features from two style images in each column, combined them and synthesized style-controlled images with combined style features and the mask image in each row.

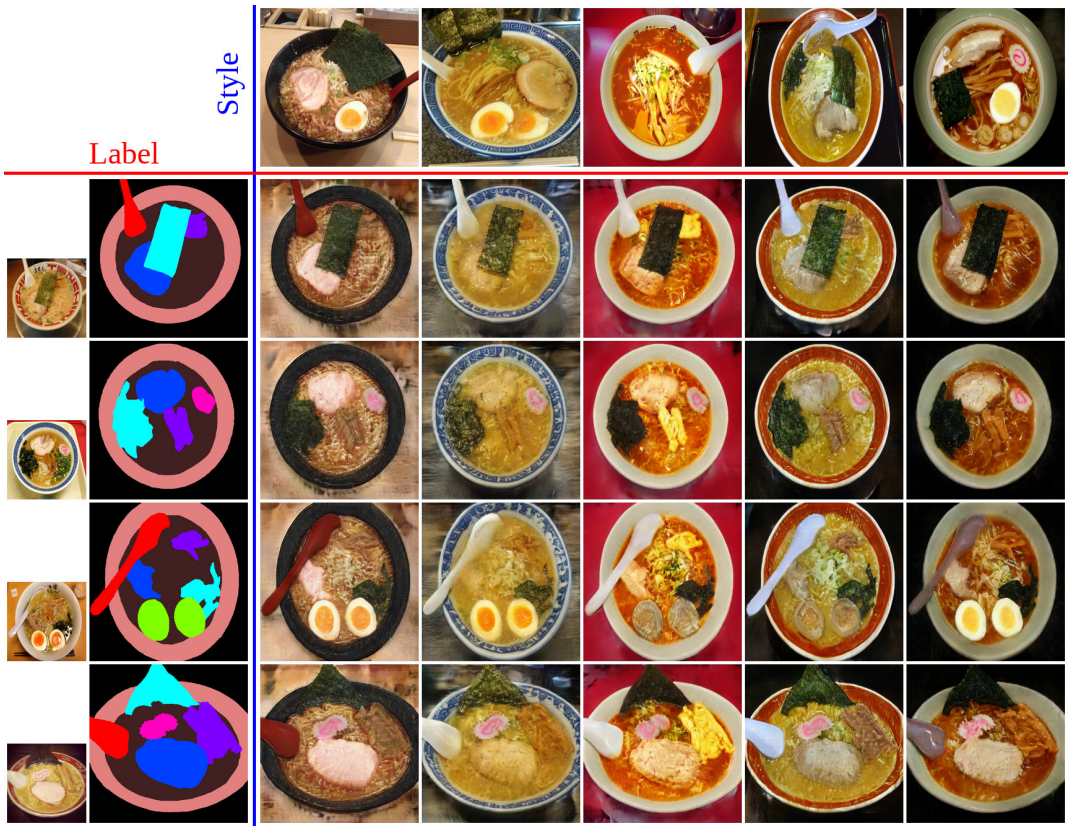
Finally, Figure 10 shows the results of image synthesis when the style features are changed gradually between two style images.

Fig. 7. Visual comparison of semantic image synthesis results on the UEC-Ramen555, CelebAMask-HQ, ADE20K and Cityscapes datasets.

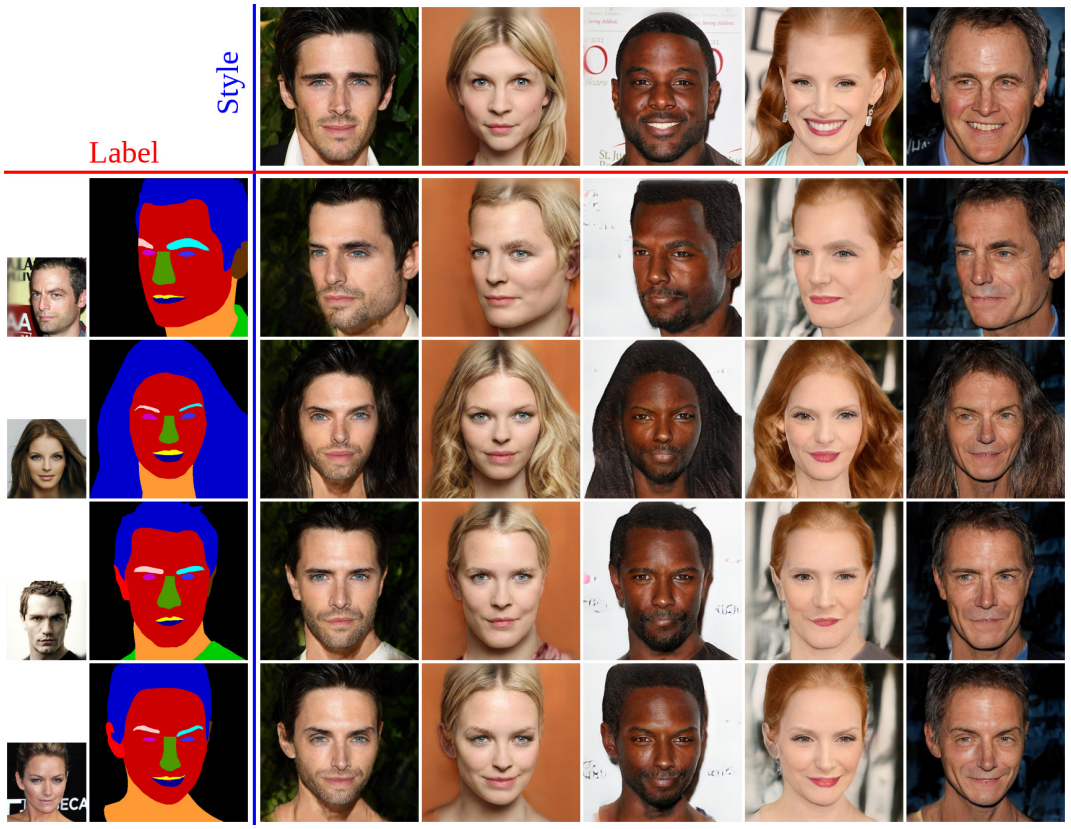
It is not easy to synthesize the style-controlled image for each element using a style image that does not correspond to the input semantic label by the existing method, SPADE. However, in our method, we have enabled style-controlled image synthesis for each mask element using mask style features extracted from different style images.

VI. Conclusions

We have proposed a novel style-controlled image synthesis method using a semantic segmentation mask by extracting style features from style images with a mask style encoder. Our



(a) UEC-Ramen555

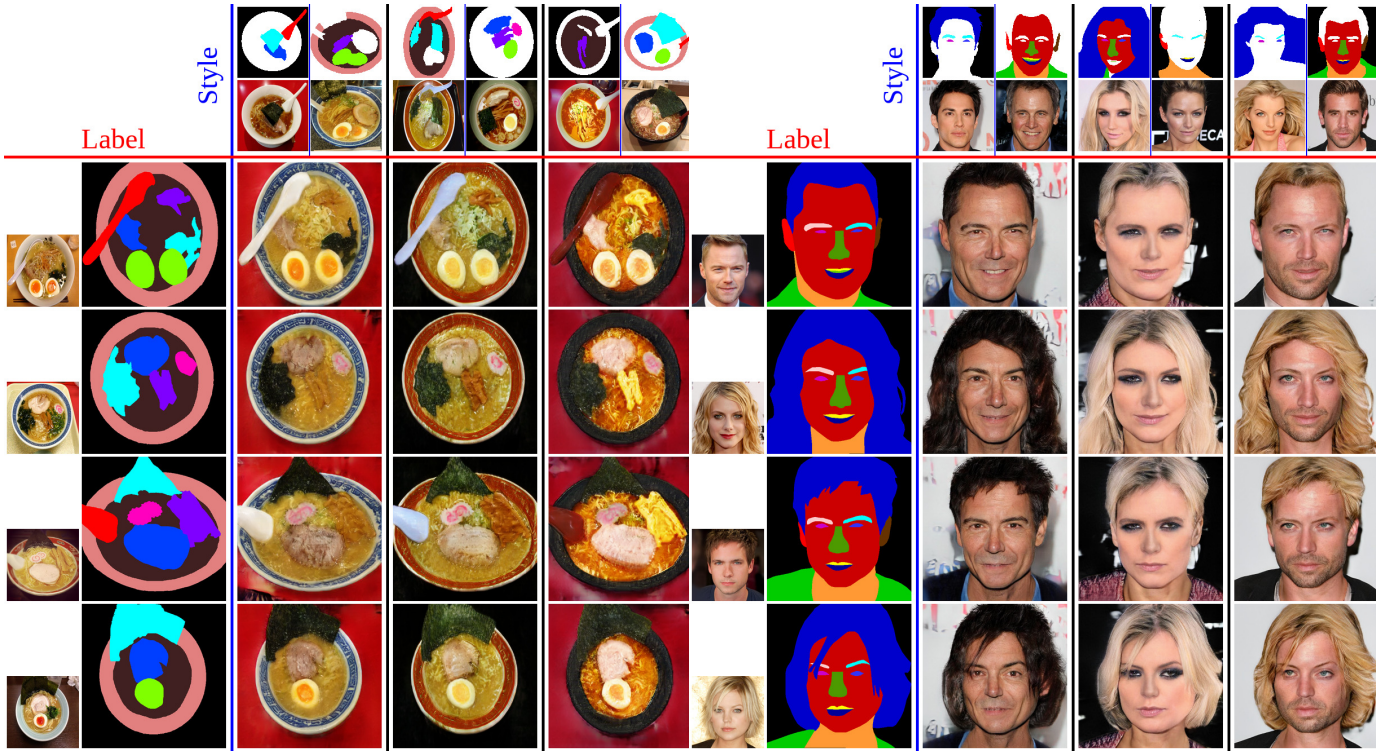


(b) CelebAMask-HQ

Fig. 8. Results of style-controlled image synthesis using various style images on the UEC-Ramen555 and CelebAMask-HQ datasets.

TABLE I
QUANTITATIVE COMPARISON USING FID SCORES FOR EACH DATASET.

Method	UEC-Ramen555	CelebAMask-HQ	ADE20K	Cityscapes
SPADE	**67.76**	34.66	29.25	**45.90**
Ours	72.10	**13.55**	**25.12**	48.98



(a) UEC-Ramen555 (b) CelebAMask-HQ

Fig. 9. Results of style-controlled image synthesis using multiple style images for each element on the UEC-Ramen555 and CelebAMask-HQ datasets.

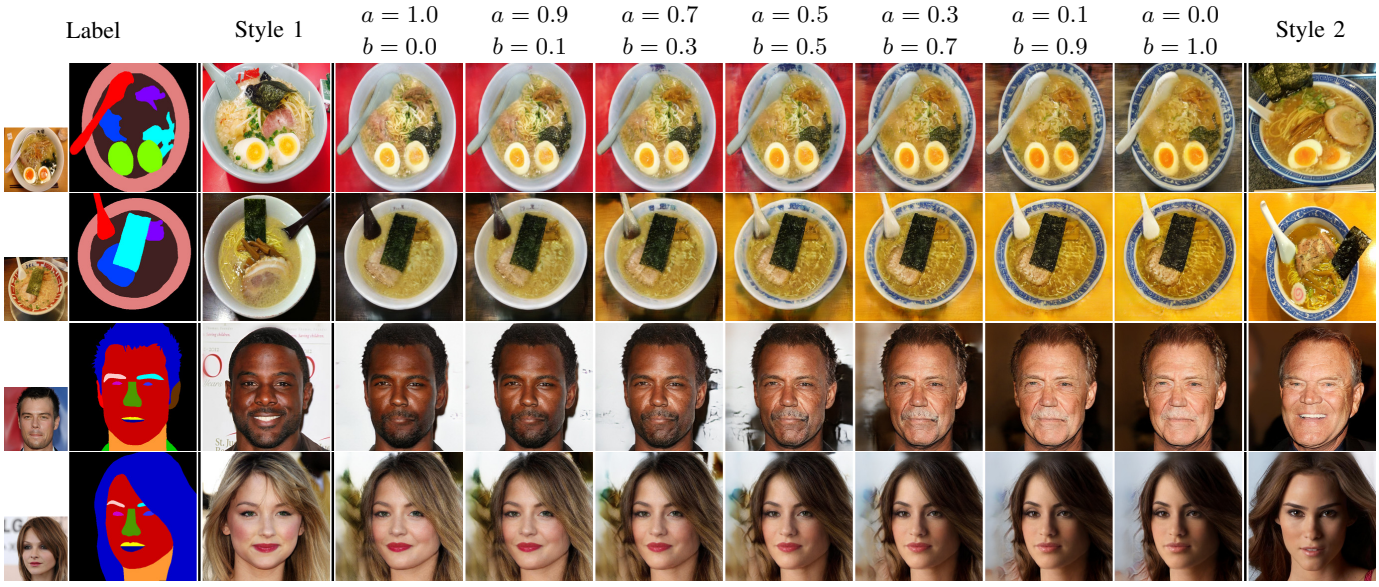


Fig. 10. Morphing results of style translation between two style images on the UEC-Ramen555 and CelebAMask-HQ datasets. a and b represent the proportion of style image 1 and style image 2, respectively.

method enables style-controlled image synthesis by concatenating the style features extracted from the mask style encoder with an input semantic mask in the SPADE-based network. The experiments on image synthesis using various datasets have shown that our method can synthesize images with equivalent quality and better style control than the existing methods. In future work, we plan to extract more detailed style features and to perform more detailed style-controlled image synthesis.

Acknowledgements: This work was supported by JSPS KAKENHI Grant Number 15H05915, 17H01745, 17H06100 and 19H04929.

REFERENCES

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[2] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2017.

[3] Q. Chen and V. Koltun, "Photographic image synthesis with cascaded refinement networks," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2017.

[4] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2018.

[5] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2019.

[6] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Proc. International Conference on Learning Representations (ICLR)*, 2016.

[7] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2017.

[8] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. International Conference on Machine Learning (ICML)*, 2017.

[9] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," in *Proc. International Conference on Learning Representations (ICLR)*, 2018.

[10] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2019.

[11] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. Efros, "Context encoders:feature learning by inpainting," in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2016.

[12] U. Demir and G. Unal, "Patch-based image inpainting with generative adversarial networks," *arXiv preprint arXiv:1803.07422*, 2018.

[13] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2018.

[14] ——, "Free-form image inpainting with gated convolution," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019.

[15] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, "Generative visual manipulation on the natural image manifold," in *Proc. European Conference on Computer Vision (ECCV)*, 2016.

[16] D. Bau, H. Strobelt, W. Peebles, J. Wulff, B. Zhou, J.-Y. Zhu, and A. Torralba, "Semantic photo manipulation with a generative image prior," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, 2019.

[17] D. Bau, J.-Y. Zhu, H. Strobelt, B. Zhou, J. B. Tenenbaum, W. T. Freeman, and A. Torralba, "Gan dissection: Visualizing and understanding generative adversarial networks," in *Proc. International Conference on Learning Representations (ICLR)*, 2019.

[18] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2017.

[19] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv:1411.1784*, 2014.

[20] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier gans," in *Proc. International Conference on Machine Learning (ICML)*, 2017.

[21] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," in *Proc. International Conference on Machine Learning (ICML)*, 2016.

[22] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2017.

[23] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2017.

[24] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman, "Toward multimodal image-to-image translation," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[25] X. Qi, Q. Chen, J. Jia, and V. Koltun, "Semi-parametric image synthesis," in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2018.

[26] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," in *Proc. International Conference on Machine Learning (ICML)*, 2017.

[27] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[28] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2018.

[29] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *Proc. European Conference on Computer Vision (ECCV)*, 2018.

[30] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. Singh, and M.-H. Yang, "Diverse image-to-image translation via disentangled representations," in *Proc. European Conference on Computer Vision (ECCV)*, 2018.

[31] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. International Conference on Medical image computing and computer-assisted intervention (MICCAI)*, 2015.

[32] P. Zhu, R. Abdal, Y. Qin, and P. Wonka, "SEAN: image synthesis with semantic region-adaptive normalization," in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2020.

[33] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *Proc. International Conference on Learning Representations (ICLR)*, 2018.

[34] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," *arXiv preprint arXiv:1805.08318*, 2018.

[35] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. International Conference on Learning Representations (ICLR)*, 2015.

[36] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. International Conference on Learning Representations (ICLR)*, 2015.

[38] J. Cho, W. Shimoda, and K. Yanai, "Ramen as you like: Sketch-based food image generation and editing," in *Proc. ACM International Conference Multimedia (MM)*, 2019.

[39] C.-H. Lee, Z. Liu, L. Wu, and P. Luo, "Maskgan: Towards diverse and interactive facial image manipulation," in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2020.

[40] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2015.

[41] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2017.

[42] 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 *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2016.

[43] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2016.