Multi-Style Transfer Generative Adversarial Network for Text Images

Honghui Yuan Keiji Yanai
Department of Informatics, The University of Electro-Communications, Tokyo, Japan
{yuan-h, yanai}@mm.inf.uec.ac.jp

***Abstract*—In recent years, neural style transfer have shown impressive results in deep learning. In particular, for text style transfer, recent researches have successfully completed the transition from the text font domain to the text style domain. However, for text style transfer, multiple style transfer often requires learning many models, and generating multiple styles images of texts in a single model remains an unsolved problem. In this paper, we propose a multiple style transformation network for text style transfer, which can generate multiple styles of text images in a single model and control the style of texts in a simple way. The main idea is to add conditions to the transfer network so that all the styles can be trained effectively in the network, and to control the generation of each text style through the conditions. We also optimize the network so that the conditional information can be transmitted effectively in the network. The advantage of the proposed network is that multiple styles of text can be generated with only one model and that it is possible to control the generation of text styles. We have tested the proposed network on a large number of texts, and have demonstrated that it works well when generating multiple styles of text at the same time.**

***Index Terms*—font translation, style transfer, text images, multi-style, GAN**

I. Introduction

Today, artistic texts are widely used in everyday life, for example, in advertising and artwork. Style transfer [1] enables content images to be converted according to the style images. AdaIN [2] proposed a new AdaIN layer, which makes the generated style image unrestricted to only the style of the dataset and enabled Arbitrary style transformation of the image in the network. MUNIT [3] enabled unsupervised image-to-image style transformations by learning content and style separately in the network. Therefore, the existing researches related to style transformation of images have made very significant progress. As a type of image style transfer, a lot of researches have been done on text style transformation. TETGAN [4] enabled the transfer of style from text to text. Image stylization [5] enabled text style transformation through a common style image. In the field of text style transformation, the network not only needs to learn the content and texture of a large number of style images, but also needs to match style features to text features, which makes the task of the network heavy. Therefore, the text style transformation has not been able to achieve multiple styles or arbitrary style transformation like regular images.

To solve this problem, we propose a multi-style transfer network for text images to explore the possibility of achieving multiple style transformations of text in a single model. Recently, Shape-Matching GAN [6] can change text styles with just one style image, which making it possible to transform text images in multiple styles. Therefore, we used Shape-Matching GAN [6] as the base network and optimized it appropriately. Our main idea is to control the generation of each text style image by adding conditions to the network and modifying the network to ensure that the features of each style image can be learned efficiently in the network. Specifically, we use a pair of mask images and style images as the input of the network. The mask images is used as a condition to control the network to learn various style features. To allow mask images to control style generation more effectively, we supplemented the network with some SPADE (SPatially-Adaptive DE-normalization) ResBlk [7]. It allows the network to retain information about the mask images more effectively. The experimental results show that our proposed method is superior in generating multiple styles compared to the previous studies of text image style transformation, and the generation of style text images achieves the expected results. In addition, we utilized a large amount of texts to verify the effectiveness of our multi-styled network.

Our main contributions are as follows:

1) we propose a multi-style transfer network for text, which can learn multiple styles in a single model.
2) We can control the generation of various styles by using a simple method. Specifically, we can control the generation of the corresponding style of text images just by using mask images.
3) The experiments have proved that our proposed network can indeed generate effective multi-styled images and is also superior in terms of image quality.

II. Related Work

A. Style transfer

Neural image style transfer [1] is the first study of CNN-based image style transfer, which can generate high quality images of any style. Real-time style transfer and super-resolution [8] proposes to use the perceptual loss function to train the feed-forward network for image style transformation. AdaIN [2] proposes a new normalization layer. Using the mean and variance of the style image as radiometric parameters, arbitrary style transformations are achieved. Swapping autoencoder [9] uses texture swapping to achieve better style transformation effects. StyleGAN [10] proposes a new generator architecture

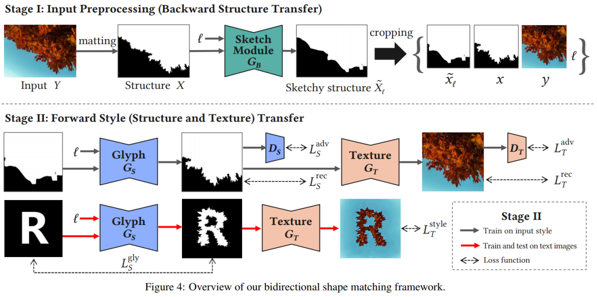

Fig. 1. The network structure of Shape-Matching GAN (cited from [6]).

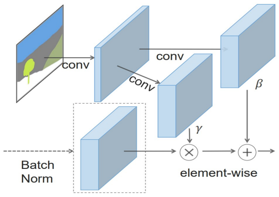

Fig. 2. The structure of SPADE layer (cited from [7]).

that can control the high-level attributes of the generated image, such as hairstyle, freckles. StyleGAN [10] can also generate high-resolution images like 1024×1024.

B. Image-to-image translation

Using the Generative Adversarial Network (GAN) [11], image-to-image translation is achieved. Pix2pix [12] enabled the generation of actual images from simple sketches or masks using paired trained data. On the other hand, CycleGAN [13] performed a transformation between two domains of images collection with unpaired data. BicycleGAN [14] enableed multiple style changes by adding VAE. UNIT [15] made the assumption that different data spaces share a potential space, enabling unsupervised transformations between different domain data sets. MUNIT [3] is an extension of UNIT [15]. The content space of the image is assumed to be shared and the style space is assumed to be independent. MUNIT [3] achieved an unsupervised multi-style transfer. StarGAN [16] is a multiple domain version of CycleGAN [13], so it enabled multi-domain conversion with only one generator and one discriminator. To achieve transformation to multiple domains, StarGAN [16] added control information about domain selection, similar to the conditional GAN [17] format. In the design of the network structure, the discriminator not only needs to learn to identify whether the sample is real or not, but also to determine which domain the real image comes from. SPADE [7] allows users to create an actual composite image from a simple image drawn by the user. The user can also choose the style of the image to be synthesized, which makes it possible to obtain a wide variety of synthesis results. SEAN [18] made improvements for SPADE [7]. A new normalization module, SEAN (semantic region-adaptive normalization) [18] was proposed. It was possible to create spatially distinct normalization parameters for each semantic region using style input images. Thus, individual control of each region of a semantic segmentation image was achieved.

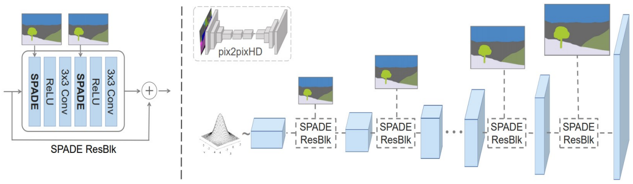

Fig. 3. The network structure of SPADE generator (cited from [7]).

C. Text font style transfer

TETGAN [4] through style transfer and style removal enabled the network to learn to decompose and recombine the content and style features of text effect images. It is useful for text image style conversion. FETGAN [19] proposed an adaptive instance-normalized font style migration for few-shots. It solved the problem of converting existing fonts to the new style while keeping the text unchanged when we have only a few new style font samples. MC-GAN [20] enabled us to change the rest of the letters to the same style based on a few letters that already exist with the style. Shape-Matching GAN [6] can transform text styles using only one style image, and can control different degrees of style. Intelligent Text Style Transfer [21] generated decorated text style images by separating, transferring and recombining decorative and basic text effects.

III. Method

In this section, we will start with a brief introduction to the basic network Shape-Matching GAN [6], then we will explain in detail what changes we have made based on Shape-Matching GAN [6].

A. Shape-Matching GAN and SPADE

The network structure of Shape-Matching GAN [6] is shown in Fig.1. The network structure is mainly divided into two stages. In Stage 1, the sketch module is used to change the style images into different degrees of deformation through the parameter $l(\in [0, 1])$. In Stage 2, there are two main parts, structure module (GS,DS) and texture module (GT,DT). The GS puts the results of the different degrees of deformation obtained in the previous stage back into the original style image to learn the structure features, while the GT learns the texture features of the style image.

The traditional batch normalization layer (BN) is easy to lose the spatial information of the mask image, so SPADE [7] proposes a new normalization layer Spatially-Adaptive Normalization to improve it. The modification lies in the difference between the calculation of γ and β. In BN, the calculation of γ and β is obtained through network training, while in Spatially-Adaptive Normalization, γ and β are calculated through mask image calculation, as shown in Fig.2. Fig.3 shows the network structure diagram of the generator in the SPADE [7].

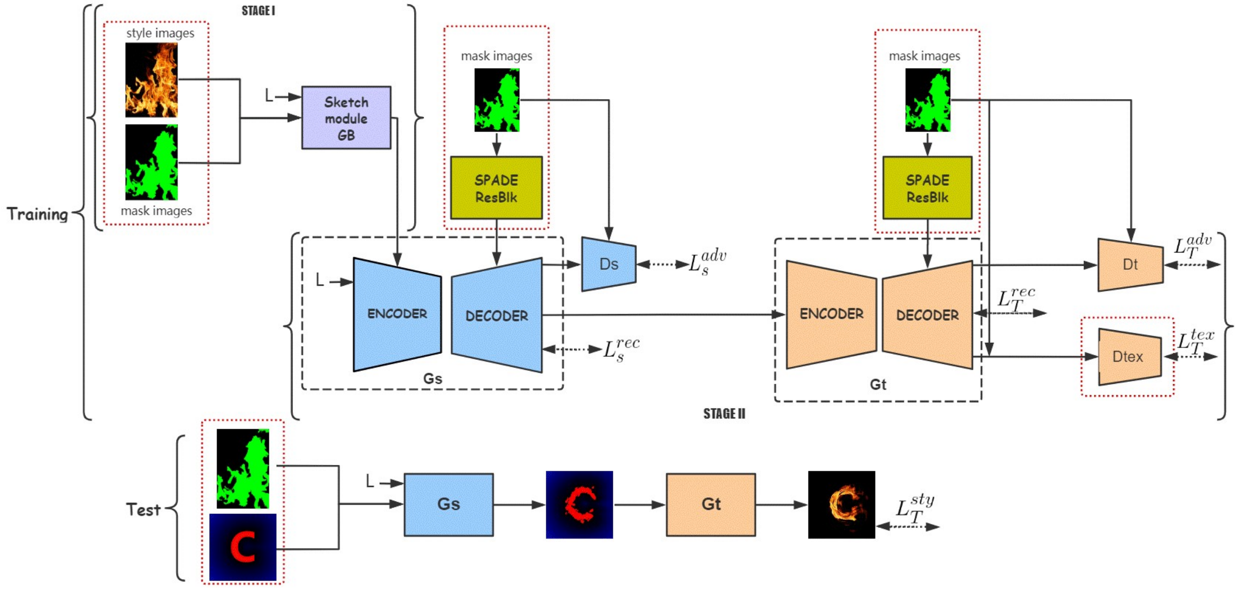

Fig. 4. The architecture of our model. We add a mask image as input on the input side to guide the generation of various styles. The SPADE ResBlk [7] were used to extract features from mask images, and input the features into the network to make mask information to be effectively passed across the network. The parts surrounded by red dotted lines are improvements we've made to the Shape MatchingGAN [6] model.

B. Proposed method

Multiple style transformations such as MUNIT [3], StyleGAN [10], and AdaIN [2], their network often needs a large number of style images for training. However, it is difficult for text image multiple style transformations. One reason is that collecting a large number of style images of various styles is difficult. Another reason is that for the task of multiple style transformation of text, it is necessary to learn both the structure of the text images, and the structure and texture of the various style images, which greatly increases the burden on the network. Therefore, in this study, since Shape-Matching GAN's [6] network requires only one style image for text style transformation, we use Shape-Matching GAN [6] as the basic network structure to achieve our multi-style text transformation task. It avoids the need to collect a large number of style images and reduces the learning burden of a multi-style network.

Our main purpose is to enable the network to effectively learn multiple styles on the basis of using only one style image, and to enable users to control the style of text images during the generation stage. So, the key question is how to achieve the learning of multiple styles in the network, and what to use to control the generation of styles.

As the recent related works, SEAN [18] and SPADE [7] achieved to generate realistic images using only a mask or a hand-drawn sketch. So, inspired by them, we would like to use mask images to control the different styles of the generated text images. Fig.4 shows the network structure that we have changed for Shape-Matching GAN [6]. The main difference is that we have added mask images as a condition to the network input in both Stage 1 and Stage 2. In addition, the normalization layer is replaced by SPADE ResBlk [7] in the GS and GT networks in Stage 2.

C. Conditional input

Conditional GAN [17] adds conditions to the GAN input, which are used to control the generation of the image. The image-to-image translation method, Pix2pix [12], is possible to generate real images from mask images or sketches. Therefore, in this study, with reference to the Conditional GAN [17], we extracted mask images for style images and put the mask images as conditions into Shape-Matching GAN's network [6]. Then, referring to SPADE [7], we extract different colors of mask images for different styles as a way to control all the styles. To allow the mask vector to control the structure and texture of the style images, we added the mask images as a condition in all the networks of the Shape-Matching GAN [6] model, and input into the network in pairs with the style images.

D. Multi-style training

If we only add a mask image to the model to achieve the generation of multiple styles of text images, the network will not be able to effectively learn the information about the mask images, which will easily lead to the integration of various styles in one image.

The mask images that guide image generation in SPADE [7] and SEAN [18] have many different colored labels. In these methods, the corresponding different parts of the generated image do not merge, and the different colored labels can generate their own part of the image very well. In the SPADE [7] method, new SPADE layer [7] was proposed. SPADE layer [7] can effectively prevent the information about mask images from being washed out in the network, making it possible for the network to effectively learn about mask image information.

Therefore, in this study, we implemented the SPADE layer in Shape-Matching GAN [6]. Specifically, we replaced the usual normalization layer in the GT and GS decoder parts of the Shape-Matching GAN [6] model with the SPADE ResBlk. Moreover, the mask of the four kinds of the style images is used as input for SPADE ResBlk.

By adding SPADE ResBlks and adding mask images as conditions, we can effectively learn multiple styles in one model, and control the generation of various style images using mask images.

E. Improving the quality of the generated images

The discriminator of Shape-Matching GAN works well when learning just one style, but it does not work as well as expected when learning multiple styles, especially for style textures. So, in this study, we add a discriminator to make the quality of the generated images better. Specifically, we add a PatchGAN discriminator [12] to our texture network by referring to the structure of the PatchGAN in the Swapping autoencoder [9].

F. Loss function

Shape-Matching GAN [6] used text images to train Sketch Module as shown in Fig.1, and we use the learned model directly, so we do not go over the loss functions for Sketch Module here. We will mainly focus on the loss functions for GS and GT. For GS networks, we add a mask image as a condition in the input of the generator and discriminator. The network GS uses reconstruction losses and adversarial losses.

In the reconstruction loss, l $(\in [0,1])$ represents the parameter that controls the degree of deformation. x represents the structural sketch obtained after binary transformation of the style image, and y represents a raw style image. $\widetilde{x}_{li}$ represents the result of style structure images with different degrees of deformation obtained from GB's network. We use a mask image as guide information to reconstruct the structure of the different style images. The reconstruction loss will restore the structure of the different degrees of images for each style to the original structure. In the adversarial loss, we added the mask images to the generator and the discriminator like Conditional GAN.

$$\begin{aligned}
\mathcal{L}_S^{rec} &= \sum_{i=1}^{N} \mathbb{E}_{x,l,mask}\left[\left\|G_s\left(\widetilde{x}_{li}, l, mask_i\right) - x_i\right\|_1\right], \quad (1)\\
\mathcal{L}_S^{adv} &= \sum_{i=1}^{N} \mathbb{E}_{x,mask}\left[\log D_S\left(x_i, mask_i\right)\right] \quad (2)\\
&+ \sum_{i=1}^{N} \mathbb{E}_{x,l,mask}\left[\log\left(1 - D_S\left(G_S\left(\widetilde{x}_{li}, l, mask_i\right)\right)\right)\right]
\end{aligned}$$

The overall GS losses are as follows:

$$\mathcal{L}_{GS} = \min_{G_S}\max_{D_S} \lambda_S^{adv}\mathcal{L}_S^{adv} + \lambda_S^{rec}\mathcal{L}_S^{rec} \quad (3)$$

The main task of the GT network is to give texture features to the structural images obtained in GS. The discriminator used in Swapping autoencoder [9] can effectively help the network learn texture features, so we added a new texture loss function $\mathcal{L}_T^{tex}$ to Shape-Matching GAN [6] based on Swapping autoencoder [9]. Thus, GT uses reconstruction losses, conditional adversarial losses, style loss, and texture loss. Style loss $\mathcal{L}_T^{sty}$ is proposed in Neural Style Transfer [1].

$$\begin{aligned}
\mathcal{L}_T^{rec} &= \sum_{i=1}^{N} \mathbb{E}_{x,y,mask}\left[\left\|G_T\left(x_i, mask_i\right) - y_i\right\|_1\right], \quad (4)\\
\mathcal{L}_T^{adv} &= \sum_{i=1}^{N} \mathbb{E}_{x,mask,y}\left[\log D_T\left(x_i, mask_i, y_i\right)\right] \quad (5)\\
&+ \sum_{i=1}^{N} \mathbb{E}_{x,l,mask}\left[\log\left(1 - D_T\left(G_T\left(x_i, mask_i\right)\right)\right)\right]
\end{aligned}$$

The overall GT losses are as follows:

$$\mathcal{L}_{GT} = \min_{G_T}\max_{D_T} \lambda_T^{adv}\mathcal{L}_T^{adv} + \lambda_T^{rec}\mathcal{L}_T^{rec} + \lambda_T^{sty}\mathcal{L}_T^{sty} + \lambda_T^{tex}\mathcal{L}_T^{tex} \quad (6)$$

$\mathcal{L}_T^{tex}$ is the loss function of Co-occurrence Patch Discriminator used in the Swapping autoencoder [9].

IV. Experiments

About the network structure, the basic structure is the same as Shape MatchingGAN [6]. The main modification is to replace the three original normalization layer of GT and GS in the decoder with SPADE ResBlks [7]. As to the question of why only three layers are replaced, because a large number of additional SPADE ResBlks [7] will increase the Calculation amounts and learning time, therefore, under the premise of ensuring the quality of the result, it is reasonable to minimize the number of the SPADE ResBlks [7]. So we have replaced only three regularization layers. The structure of the newly added discriminator is referenced to the Swapping autoencoder [9].

A. Dataset

We used 129 text images and 4 style images provided by Shape-Matching GAN's authors, which text images were only for test set. We added the corresponding mask images for the 4 style images. The text images are transformed in Shape-Matching GAN [6] using the distance function, Fig.5 shows some example images of the datasets.

B. Network training

For the Sketch Module (Fig.1), we use the existing model data of Shape-Matching GAN [6]. We mainly train on the style structure network and texture network. In the training process, we input the style images and the corresponding mask images into the network in pairs. In the testing stage, we input the selected text image and style mask image to generate the corresponding style text image.

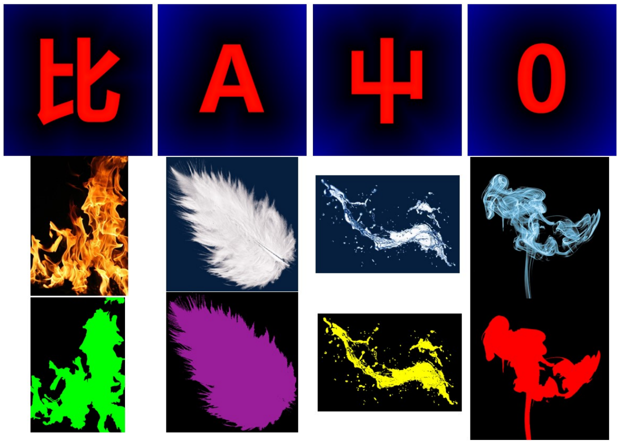

Fig. 5. Examples of dataset on styles, corresponding masks and font images.

C. Results of the experiments

The most similar methods to our method is multi-style transfer methods. However, MUNIT [3] and other methods require a large number of style images to achieve the style transformation. As a multi-domain method, StarGAN [16] also requires a large number of images for each of the domains to learn. Therefore, since only one style image is needed for this study, it is difficult to compare with these methods. Therefore, we chose arbitrary style transformations for the comparison test. AdaIN [2] and neural style transfer [1] performed well in achieving arbitrary style transformations, so that we made a qualitative comparison with these two methods and the baseline, Shape-MatchingGAN [6].

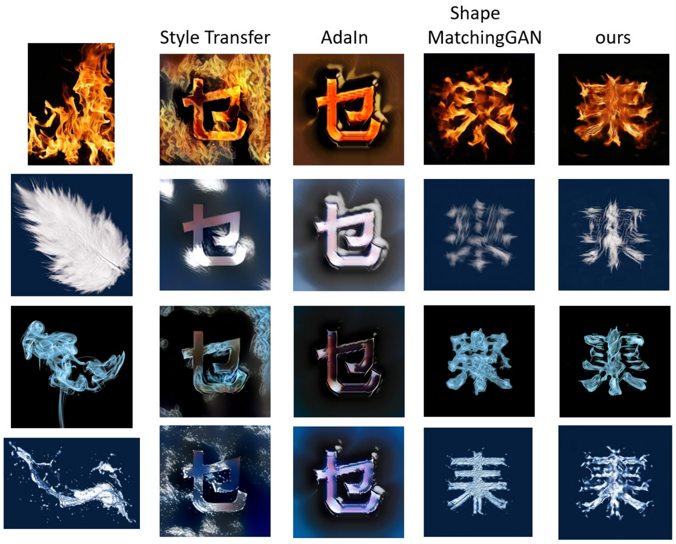

Fig. 6. Comparison between the results by our method and other baseline methods. It can be seen that AdaIN and Style transfer are not very effective in terms of text style transformation. Our results and the results of Shape-Matching GAN [6] have achieved good results in text style transfer.

In AdaIN [2] and style transfer [1], we use the learned model data and input text images and style images to run experiments. In comparison with Shape-Matching GAN [6], our approach requires learning just one model to achieve four styles of transformation. On the other hand, Shape-Matching GAN [6] learns the four models separately to perform the corresponding style transformations. The four styles of text images generated at once in our network were compared with the results generated by Shape-Matching GAN [6] separately, and with the results of the other two methods. The comparison results are shown in Fig.6.

In AdaIN [2] and style transfer [1], there is no specific learning for text style structure, So the results were not well. Because our approach and Shape MatchingGAN are specific to text, these two methods perform better for text style transformations. In a comparison with the basic method Shape-MatchingGAN [6], the multiple style results we generated in one model were as good as or better than the results obtained from Shape-Matching GAN [6] that was trained separately. It also can be seen that our approach successfully achieves multiple styles of text transformations in one model and the results have stylistic features in both structure and texture. The results of our model are shown in Fig.7.

We performed an ablation study on the text images. Fig.8 shows the results of the ablation studies. When only adding mask images to the network for input, the network does not discriminate the individual styles well, producing a result where the styles merge with each other. In the absence of a texture discriminator, the learning of style materials is not satisfactory. The results of the full model show that our network is effective in learning multiple styles and produces good results.

D. User study

We conducted a user study on the Amazon Mechanical Turk on all the four styles of images. The test set used in this study includes three types of images: 4 style images, 25 text images, and 50 transformed text images. In this study, on the test set, we conducted user study of 42 users. For the test, we show users the style image, the original text image, and the two transformed result images. One of the two result images is from our results, and the other is from the results of Shape-Matching GAN [6]. The users need to select the better image among the two given results for the following questions: *"Which of the following not only ensures the readability of the text, but also shows the style characteristics of the style image well ?"*

We counted user votes for each style. The results are shown in Fig.9. They showed that our proposed method outperformed the baseline method regarding the three styles, and it was identical regarding the other one style, "fire style."

V. Conclusions

In this study, we proposed a multi-style transfer network for text. By adding masks for style images and reform the network structure, we achieve the task of multiple style transformations for text images in one model. In addition, we can also control the generation of various styles of text images in the generation stage. The results show that we have achieved

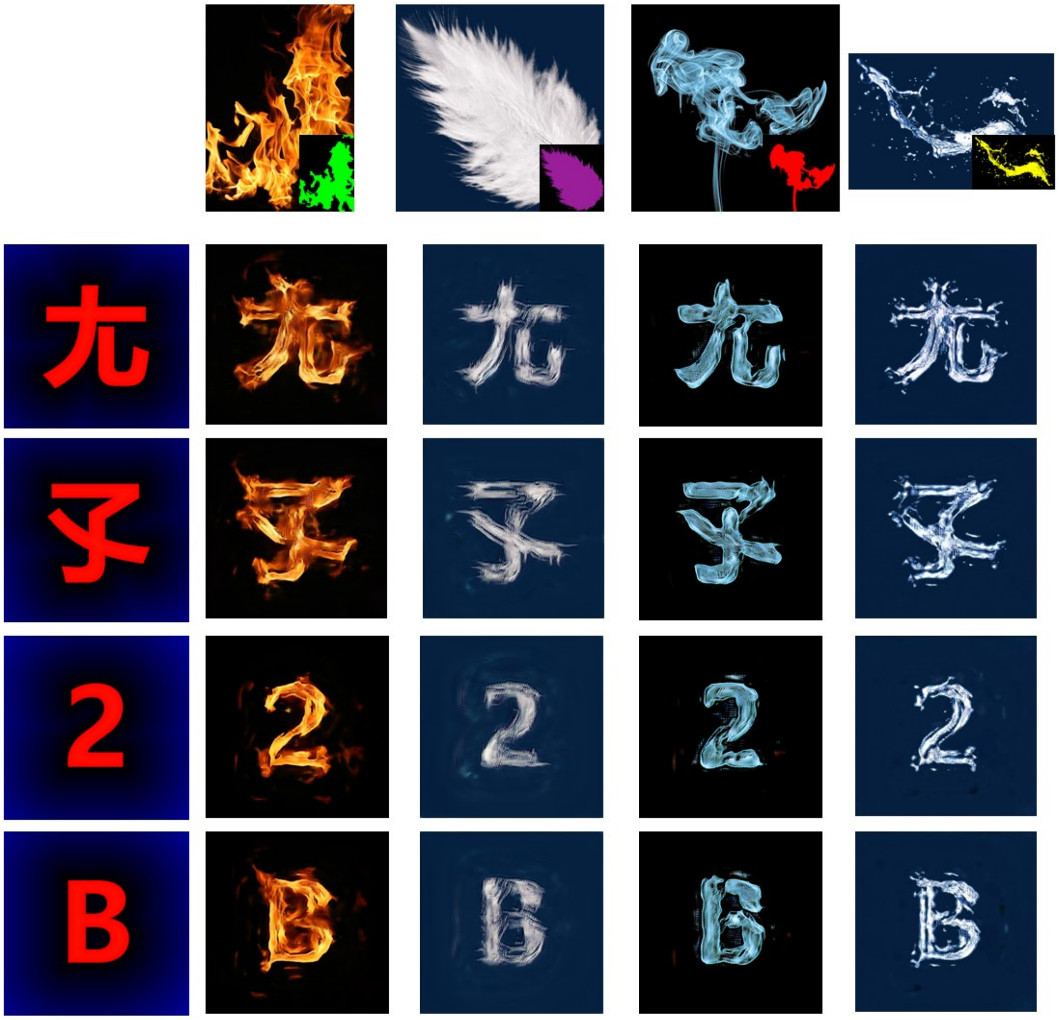

Fig. 7. The results of our model. The left side is the input text images, and the top is the style images and their mask images. The results show that our model has successfully implemented multi-style text transformation.

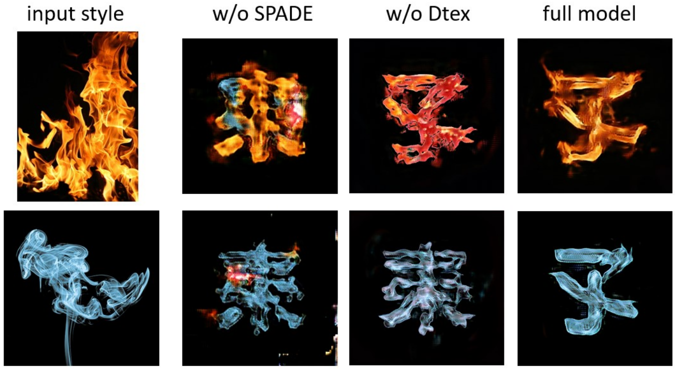

Fig. 8. The results of ablation study.

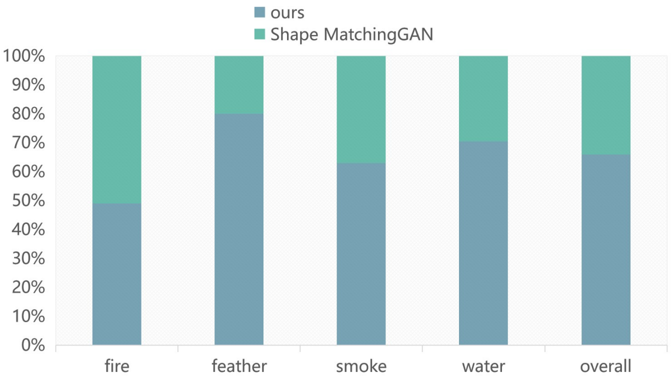

Fig. 9. The results of the user study.

a good effect on the generated style images based on the effective transformation of multiple text styles.

For future work, at this stage of the study, we have achieved four style transformations. In addition, through further research, we hope to achieve more style transformations or even arbitrary style transformations just like arbitrary style transfer methods for generic images.

References

[1] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proc.of IEEE Computer Vision and Pattern Recognition*, 2016, pp. 2414–2423.

[2] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proc.of IEEE Computer Vision and Pattern Recognition*, 2017, pp. 1501–1510.

[3] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *Proc.of European Conference on Computer Vision*, 2018, pp. 172–189.

[4] S. Yang, J. Liu, W. Wang, and Z. Guo, "Tet-gan: Text effects transfer via stylization and destylization," in *Proc.of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 1238–1245.

[5] S. Yang, J. Liu, W. Yang, and Z. Guo, "Context-aware text-based binary image stylization and synthesis," *IEEE Transactions on Image Processing*, vol. 28, no. 2, pp. 952–964, 2018.

[6] S. Yang, Z. Wang, Z. Wang, N. Xu, J. Liu, and Z. Guo, "Controllable artistic text style transfer via shape-matching GAN," in *Proc.of IEEE Computer Vision and Pattern Recognition*, 2019, pp. 4442–4451.

[7] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proc.of IEEE Computer Vision and Pattern Recognition*, 2019, pp. 2337–2346.

[8] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc.of European Conference on Computer Vision*, 2016, pp. 694–711.

[9] T. Park, J.-Y. Zhu, O. Wang, J. Lu, E. Shechtman, A. A. Efros, and R. Zhang, "Swapping autoencoder for deep image manipulation," *arXiv preprint arXiv:2007.00653*, 2020.

[10] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc.of IEEE Computer Vision and Pattern Recognition*, 2019, pp. 4401–4410.

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," 2014, pp. 2672–2680.

[12] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc.of IEEE Computer Vision and Pattern Recognition*, 2017, pp. 1125–1134.

[13] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc.of IEEE International Conference on Computer Vision*, 2017, pp. 2223–2232.

[14] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman, "Toward multimodal image-to-image translation," 2017, pp. 465–476.

[15] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," 2017, pp. 700–708.

[16] 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.of IEEE Computer Vision and Pattern Recognition*, 2018, pp. 8789–8797.

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

[18] P. Zhu, R. Abdal, Y. Qin, and P. Wonka, "Sean: Image synthesis with semantic region-adaptive normalization," in *Proc.of IEEE Computer Vision and Pattern Recognition*, 2020, pp. 5104–5113.

[19] W. Li, Y. He, Y. Qi, Z. Li, and Y. Tang, "FET-GAN: Font and effect transfer via k-shot adaptive instance normalization." in *Proc.of the AAAI Conference on Artificial Intelligence*, 2020, pp. 1717–1724.

[20] S. Azadi, M. Fisher, V. G. Kim, Z. Wang, E. Shechtman, and T. Darrell, "Multi-content gan for few-shot font style transfer," in *Proc.of IEEE Computer Vision and Pattern Recognition*, 2018, pp. 7564–7573.

[21] W. Wang, J. Liu, S. Yang, and Z. Guo, "Typography with decor: Intelligent text style transfer," in *Proc.of IEEE Computer Vision and Pattern Recognition*, 2019, pp. 5889–5897.